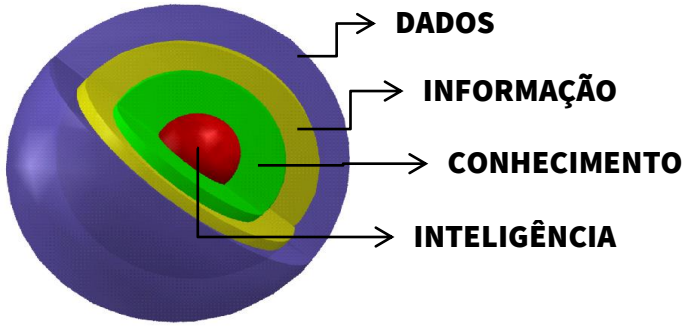
Machine Learning (Métodos supervisionados)

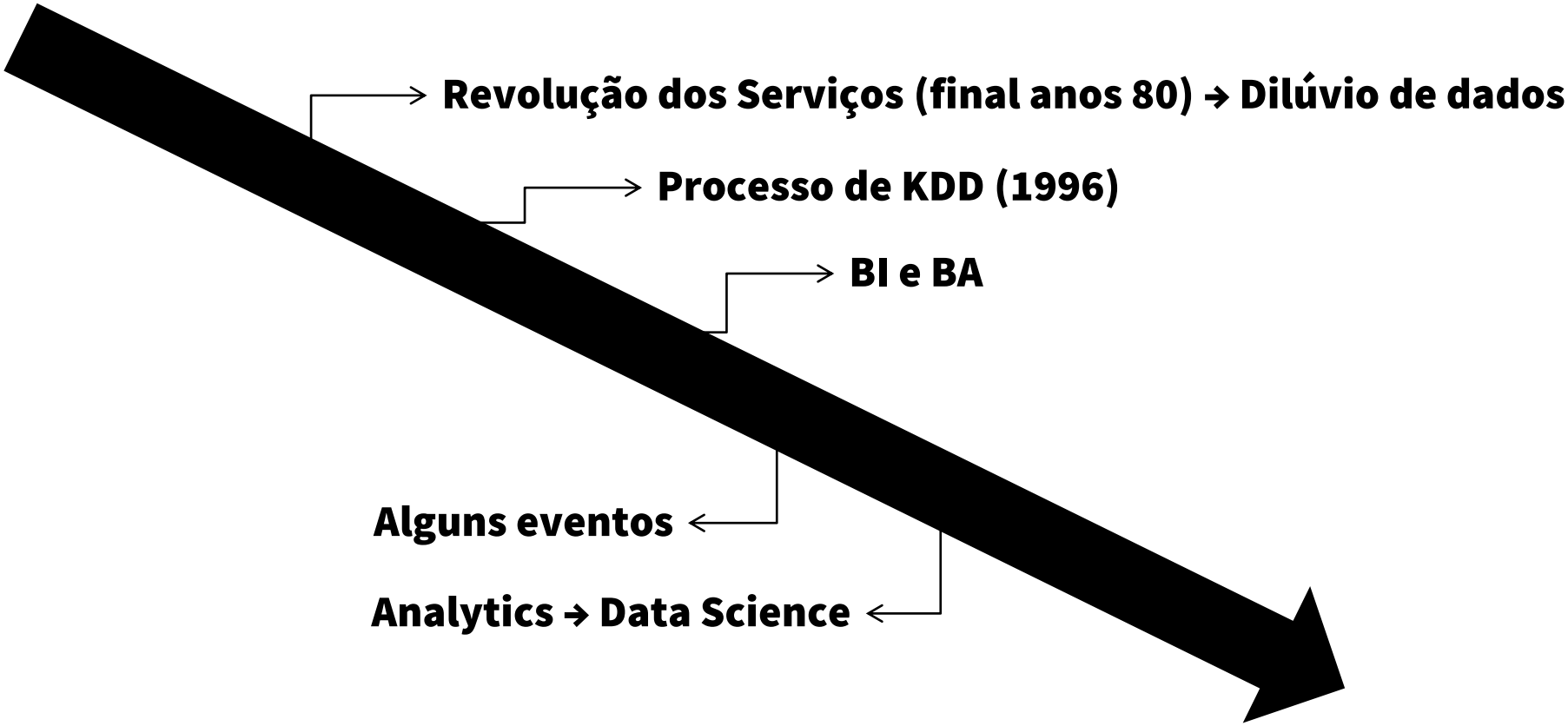


Rodrigo A. Scarpel

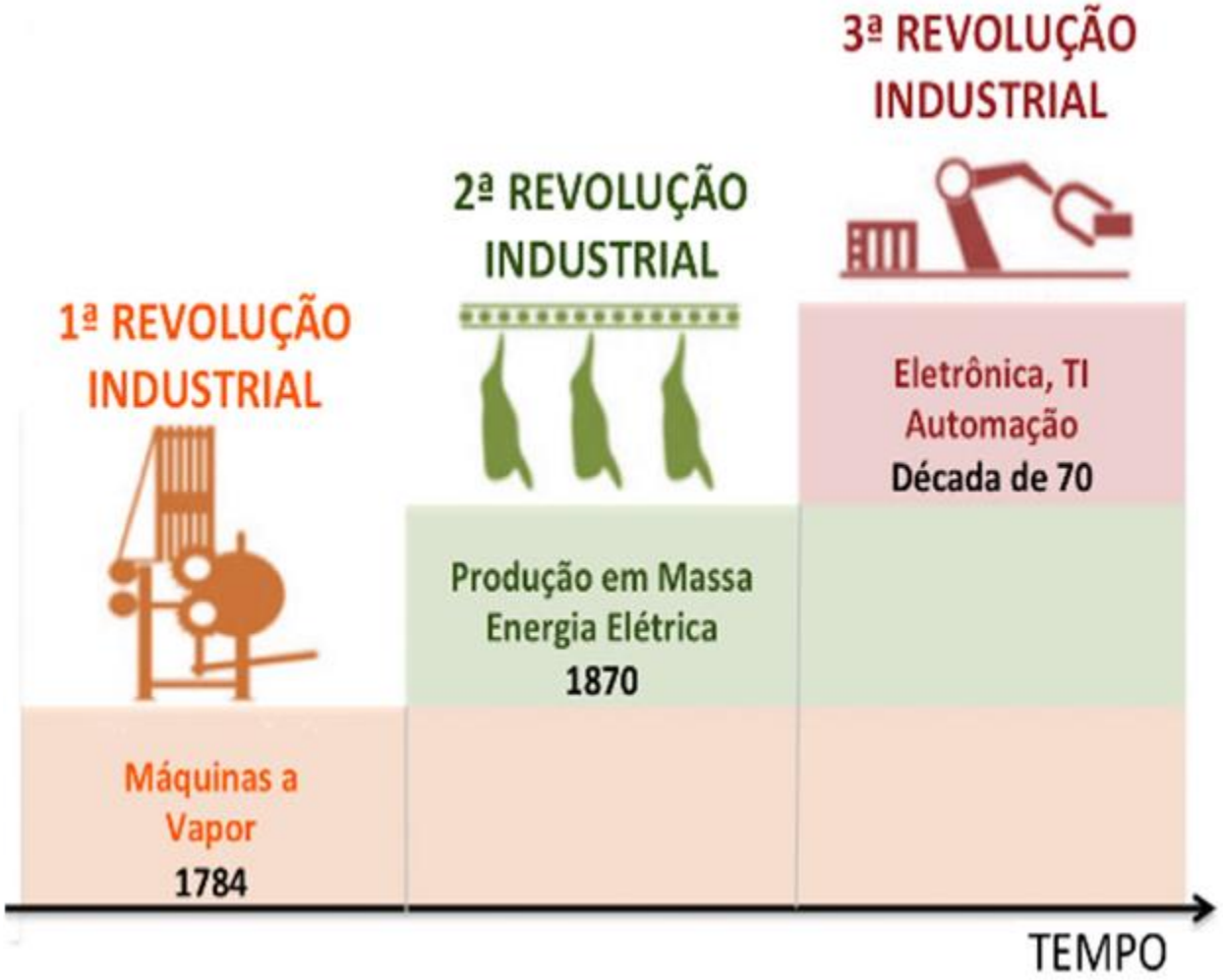
rodrigo@ita.br

www.ief.ita.br/~rodrigo

Introdução:


Revolução dos Serviços (final anos 80) → Dilúvio de dados
Processo de KDD (1996)
BI e BA
Alguns eventos
Analytics → Data Science

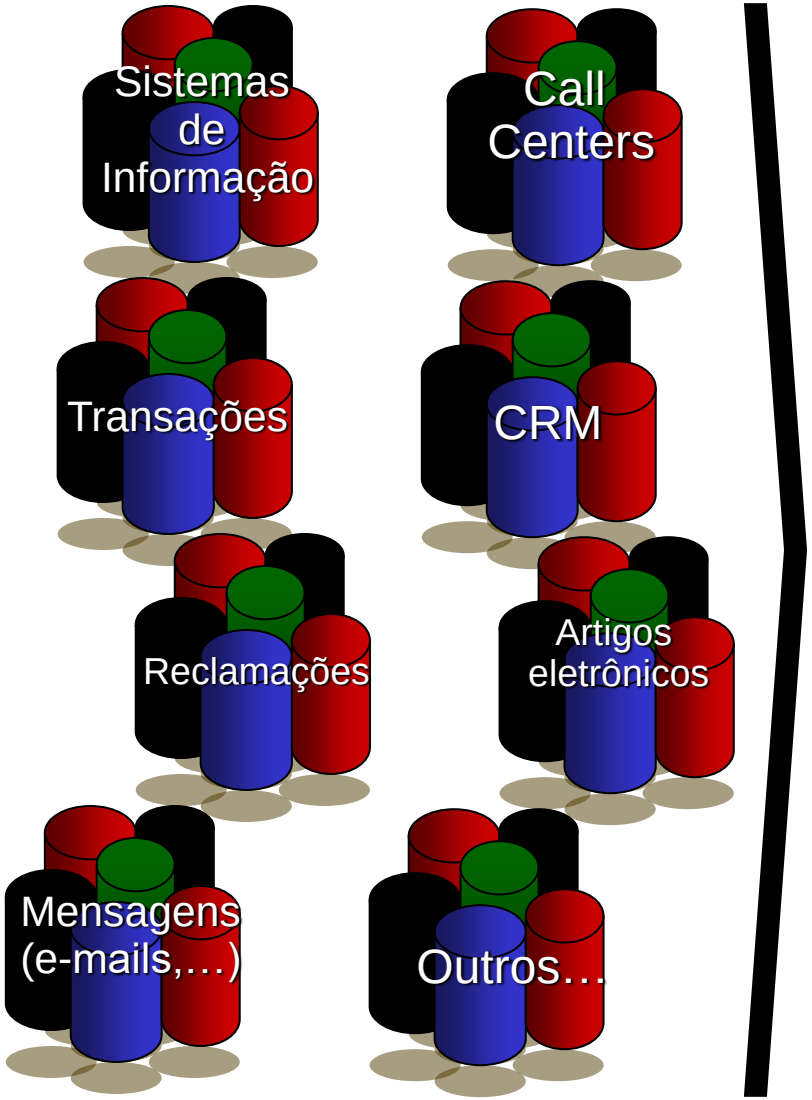
A Terceira Revolução Industrial:

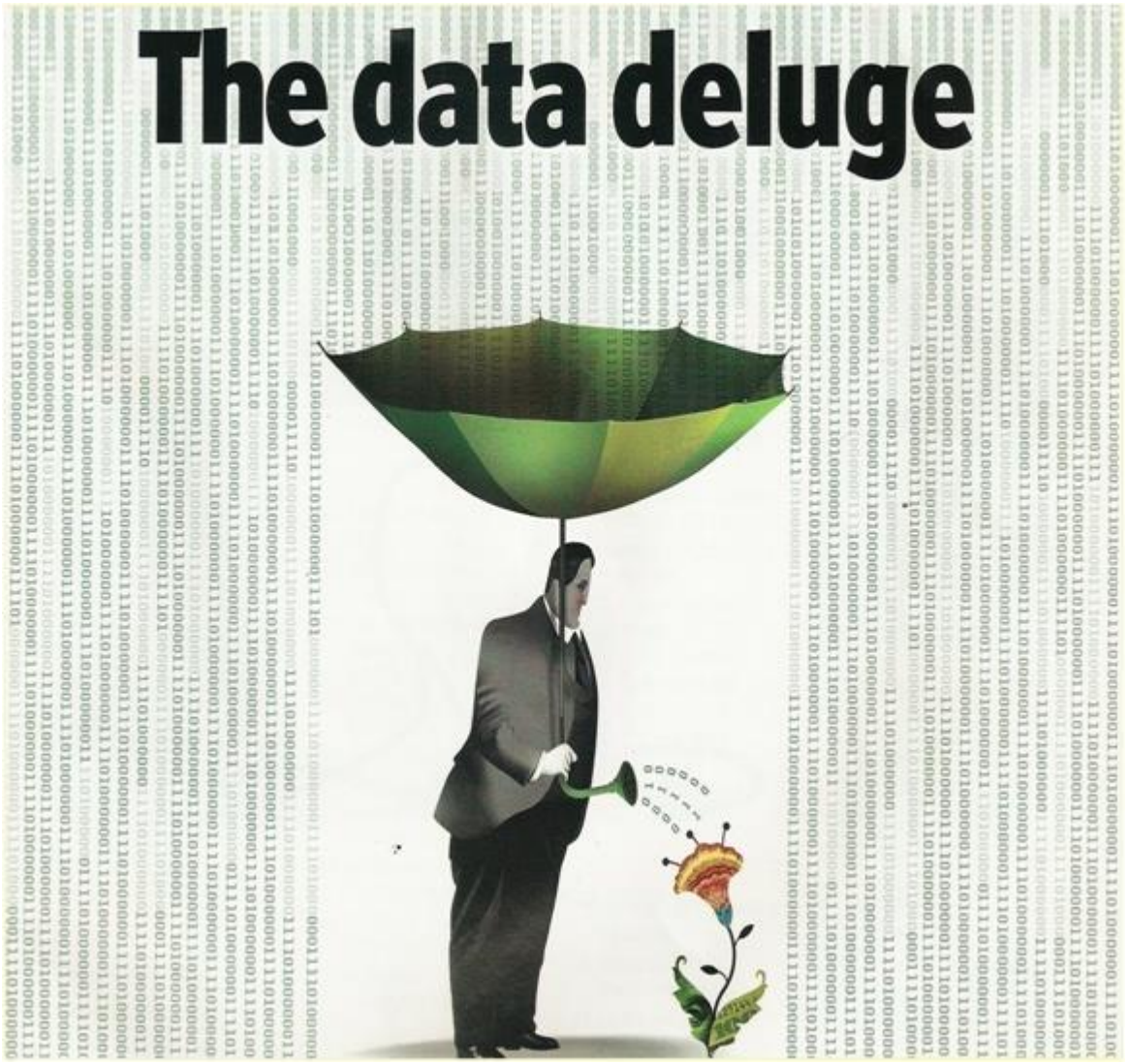


A Revolução dos Serviços (1980 a ?):

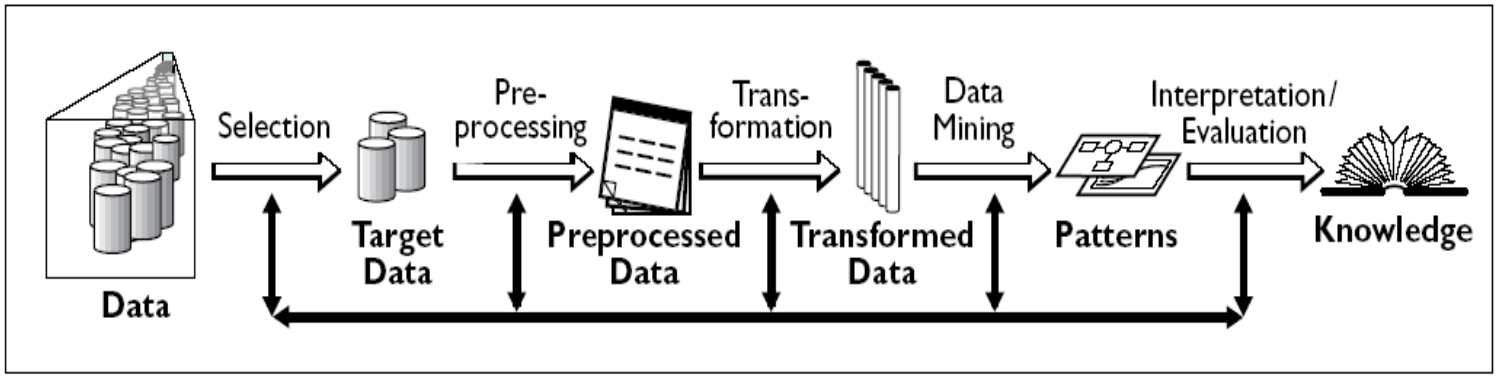
- Microinformática
- Tecnologia da informação
- Softwares
- Telecomunicações
- Setor financeiro
- Grandes varejistas
- Educação e ensino
- Internet (década de 90)

Um dilúvio de dados:

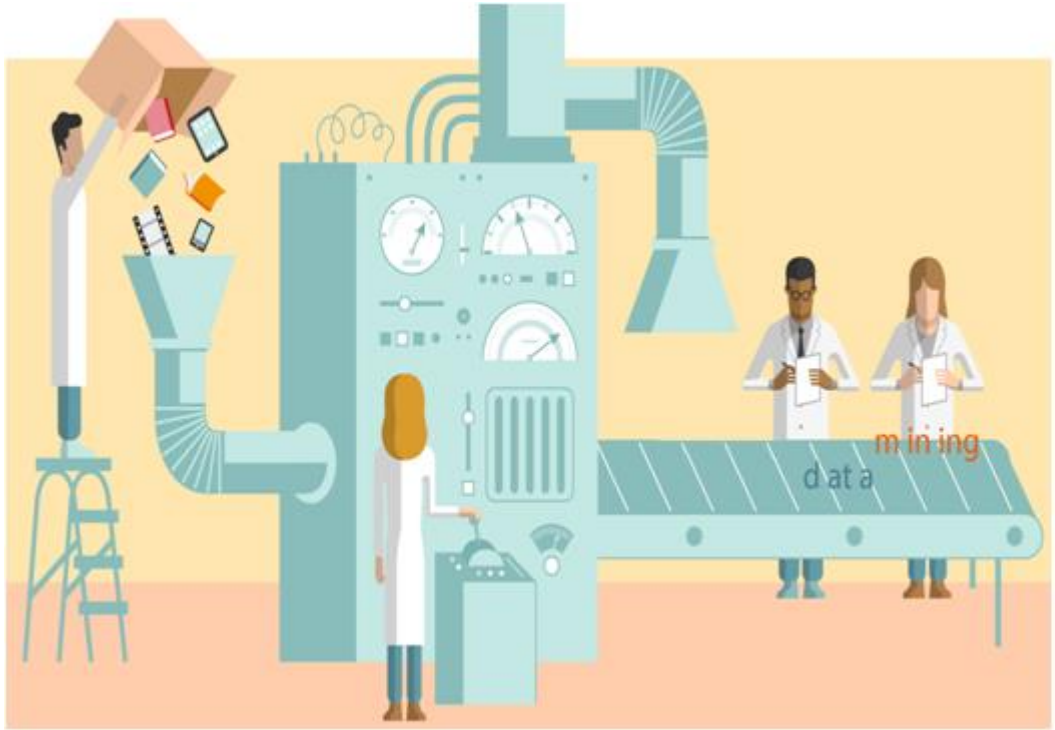




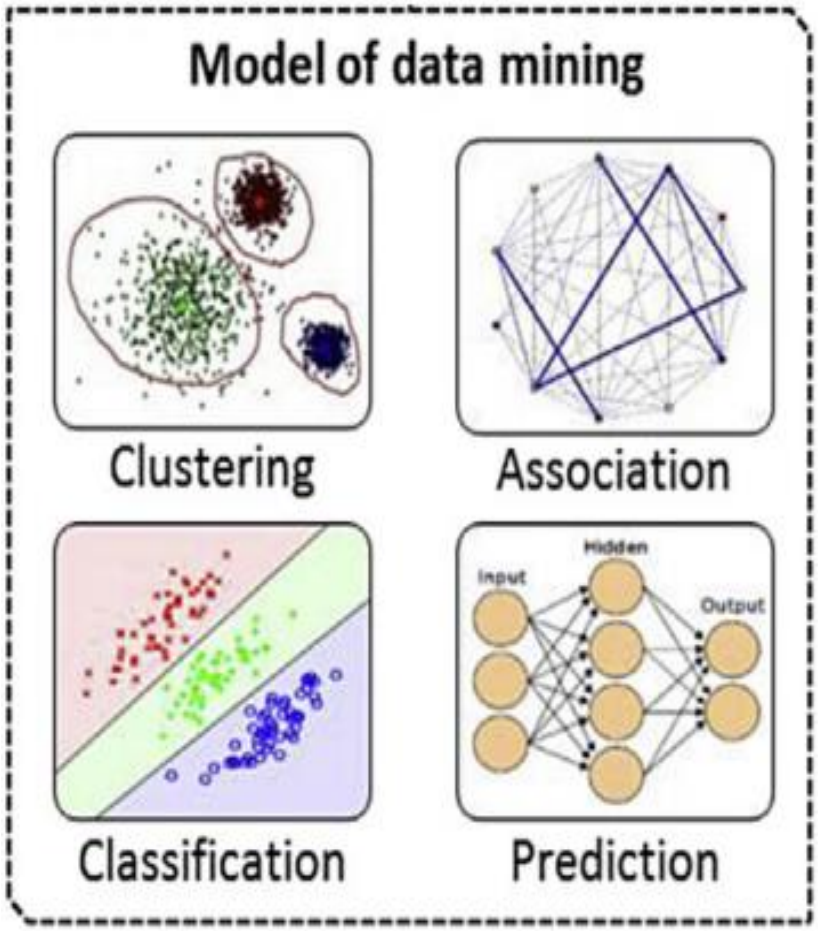
O processo de KDD:

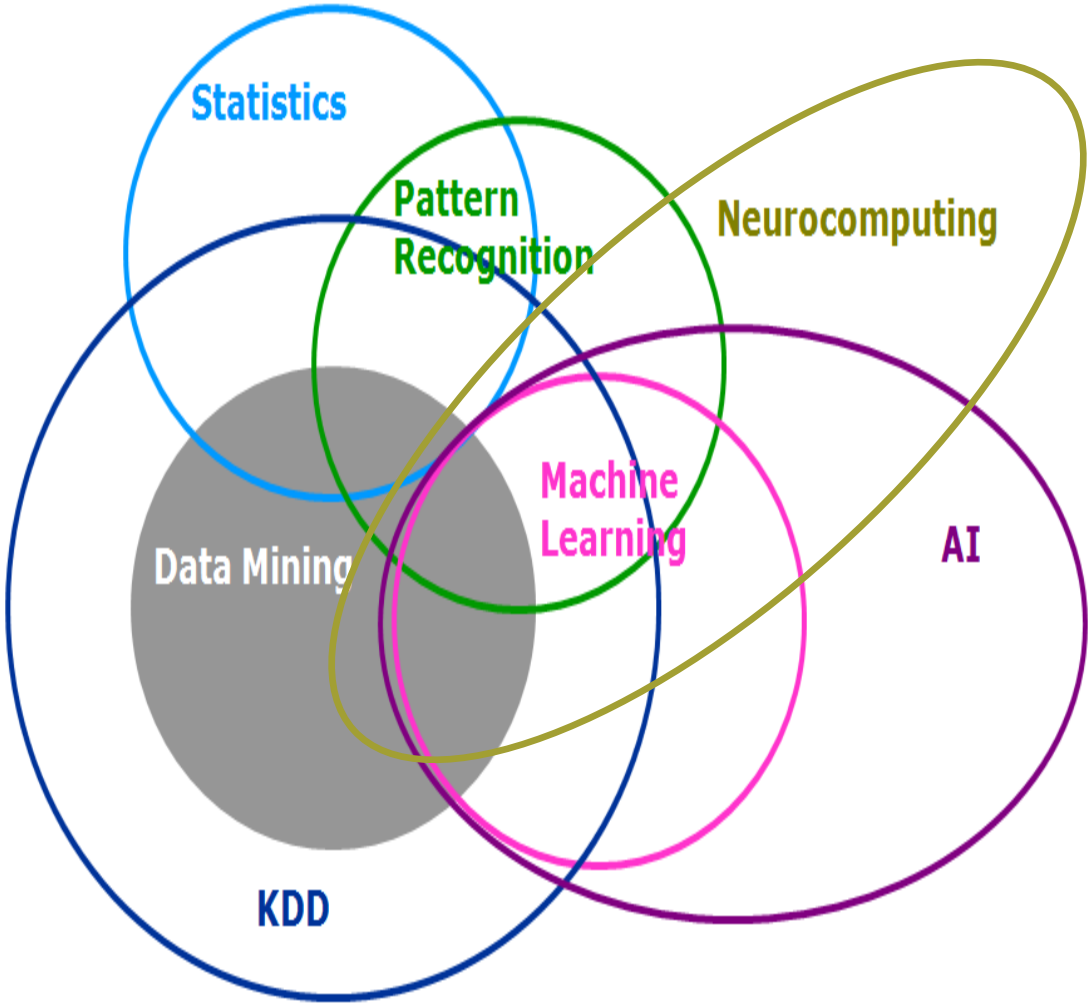


Fonte: FAYYAD, U., PIATETSKY-SHAPIRO, G., SMYTH, P. From data mining to knowledge discovery: An overview. In: Advances in Knowledge Discovery and Data Mining, AAAI Press / The MIT Press, MIT, Cambridge, Massachusetts, 1996, p.1-34.

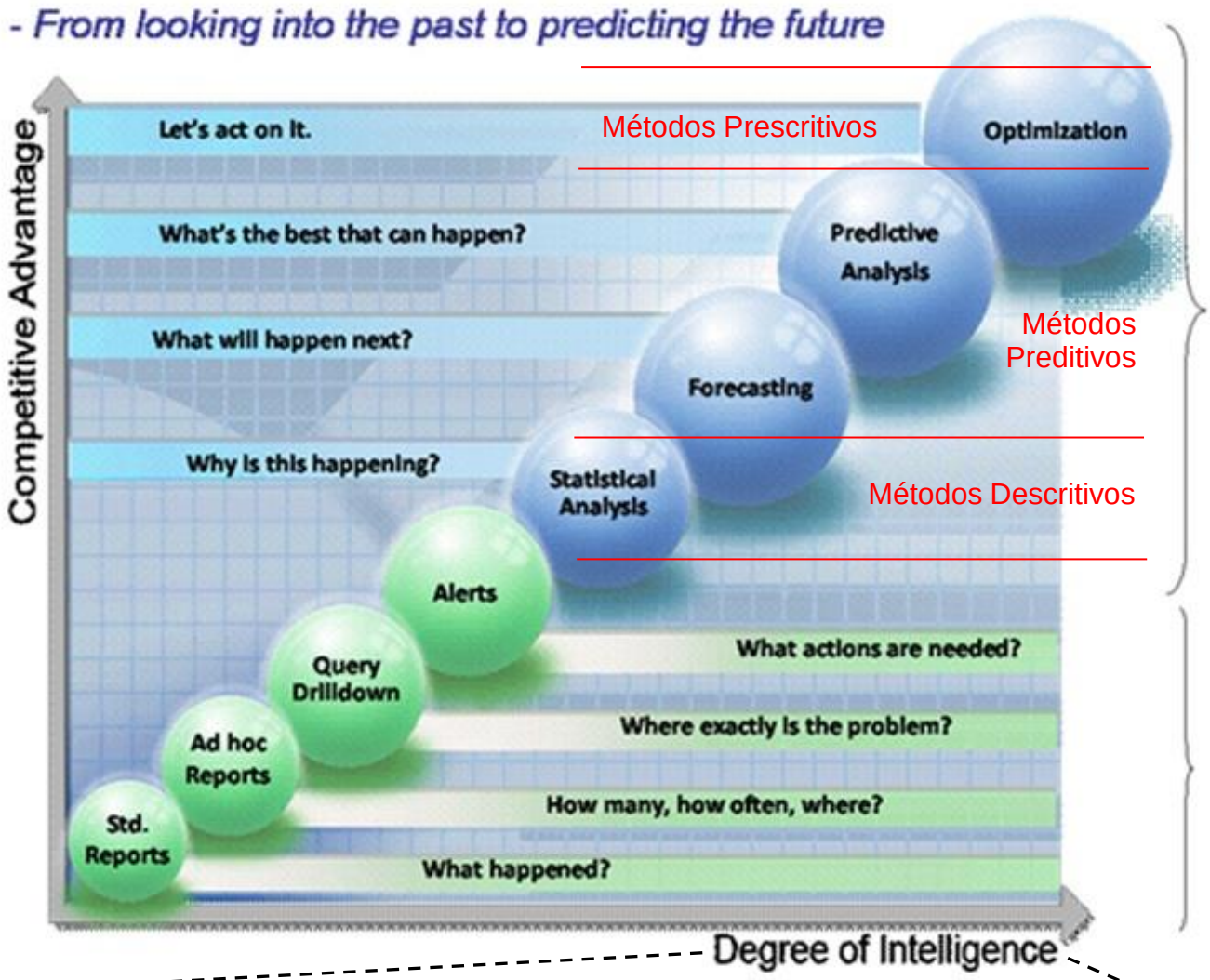


Mineração de dados:





Data Analytics:

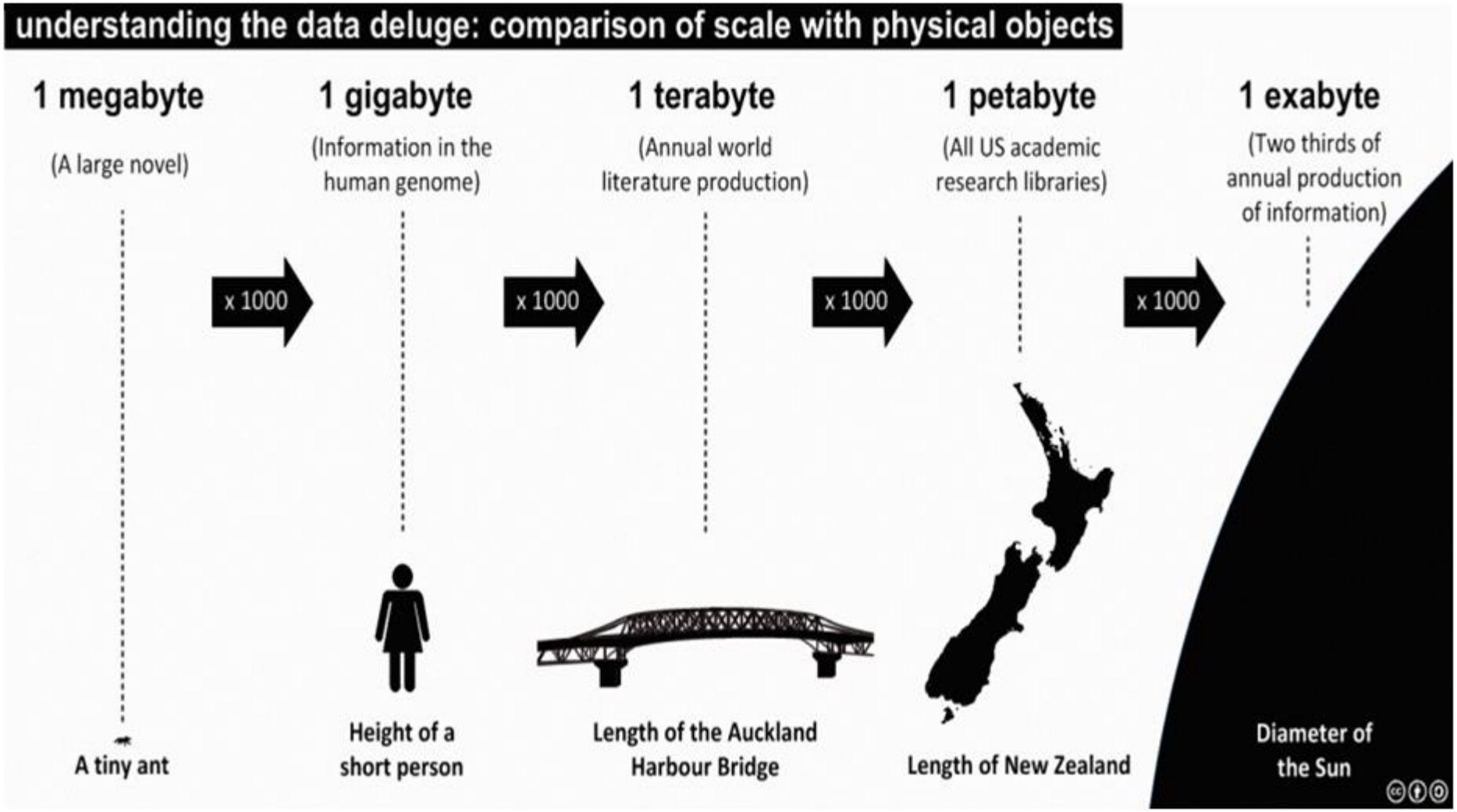


Proactive decision making ⇔ DA

Reactive decision making ⇔ DI

Dados ⟶ Informação ⟶ Conhecimento ⟶ Inteligência

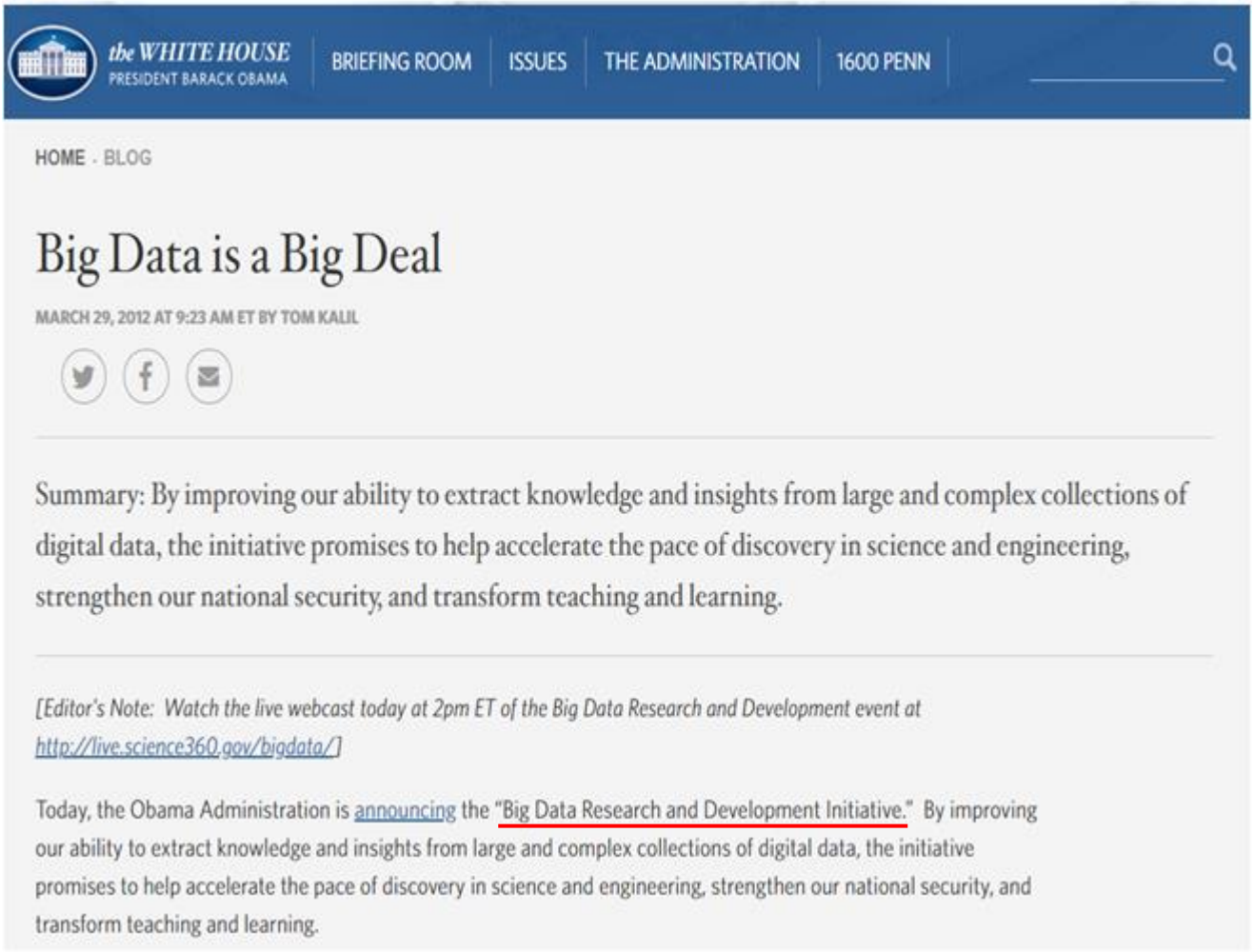
Alguns eventos:



Dilúvio de dados ⟶ Universo de dados

Fonte: http://ritholtz.com/2016/09/162347/

Alguns eventos (Big Data Initiative):



the WHITE HOUSE
PRESIDENT BARACK OBAMA

BRIEFING ROOM | ISSUES | THE ADMINISTRATION | 1600 PENN

HOME · BLOG

Big Data is a Big Deal

MARCH 29, 2012 AT 9:23 AM ET BY TOM KALIL

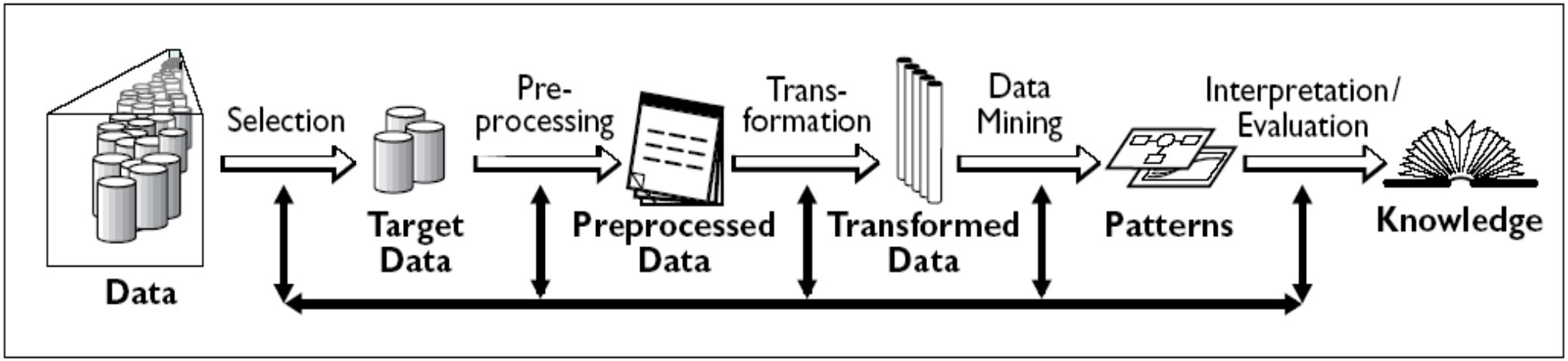
Summary: By improving our ability to extract knowledge and insights from large and complex collections of digital data, the initiative promises to help accelerate the pace of discovery in science and engineering, strengthen our national security, and transform teaching and learning.

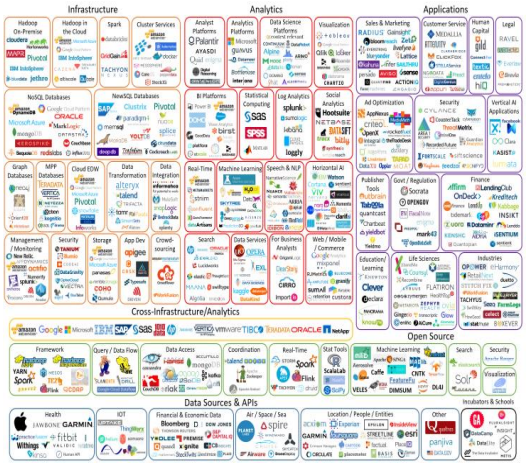
[Editor's Note: Watch the live webcast today at 2pm ET of the Big Data Research and Development event at http://live.science360.gov/bigdata/]

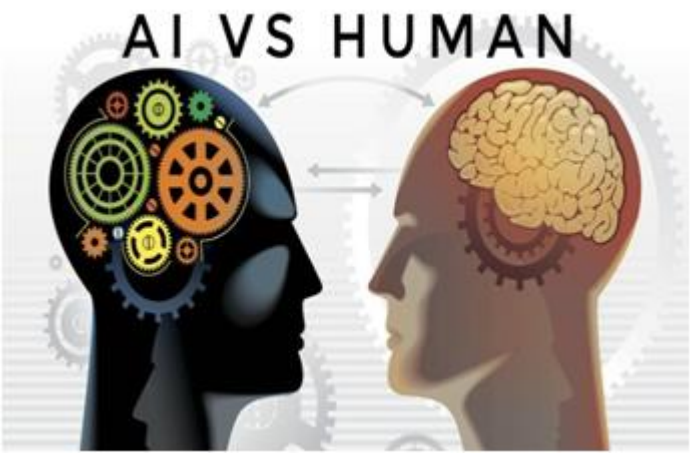
Today, the Obama Administration is announcing the "Big Data Research and Development Initiative." By improving our ability to extract knowledge and insights from large and complex collections of digital data, the initiative promises to help accelerate the pace of discovery in science and engineering, strengthen our national security, and transform teaching and learning.

Fonte: https://obamawhitehouse.archives.gov/blog/2012/03/29/big-data-big-deal

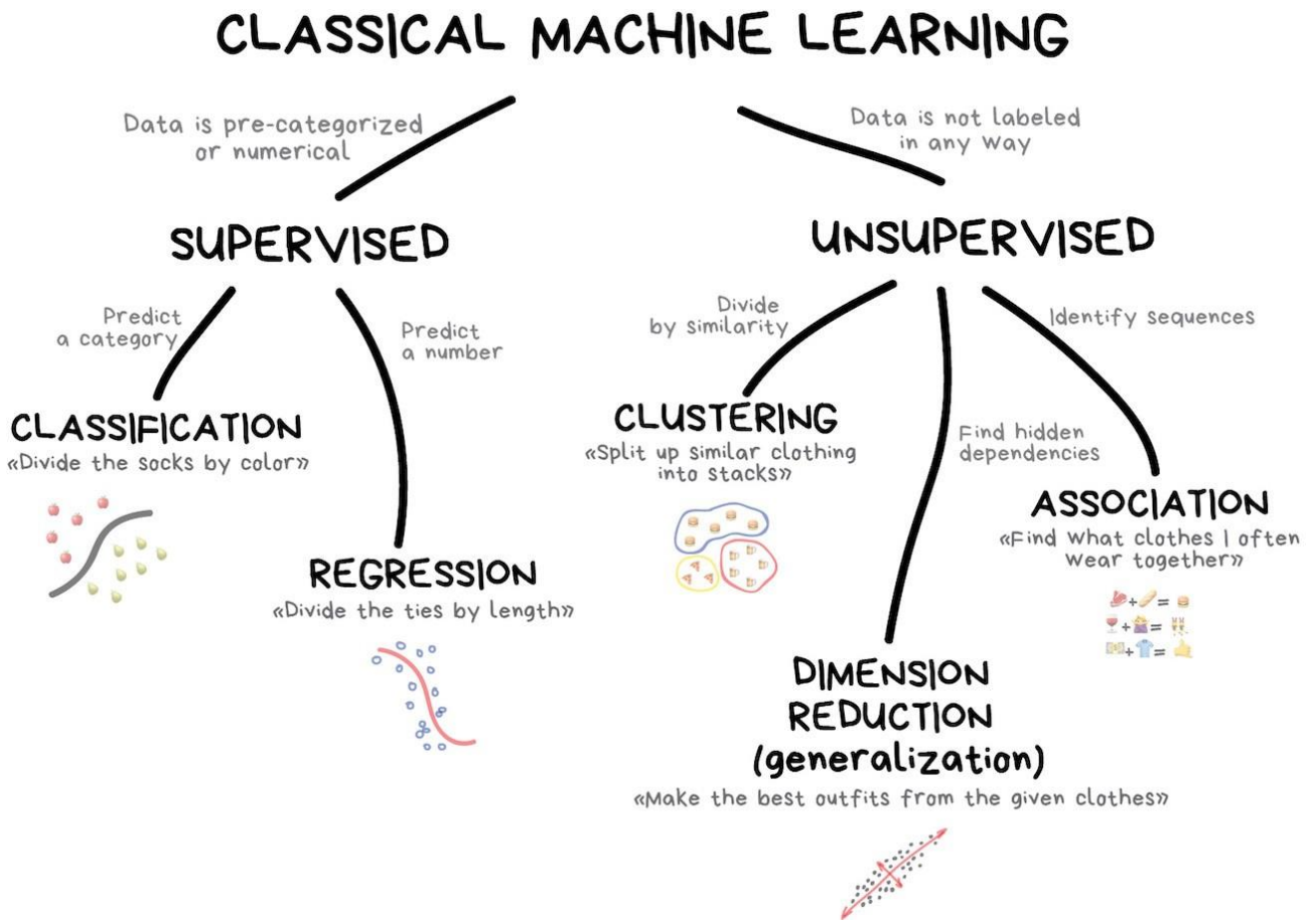
Analytics ⟶ Data Science:







Machine Learning:

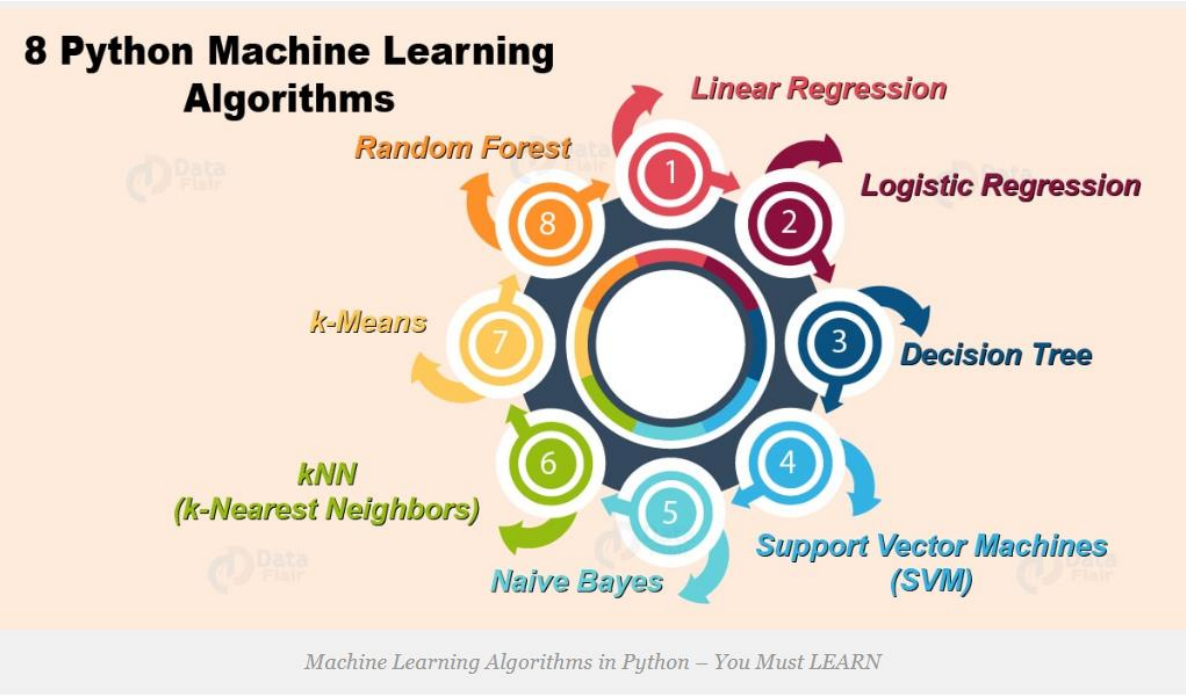


Classificação – problemas em que a variável resposta é categórica
(ex: y = fraude / não fraude
y = número 2 / número 7
y = falha / não falha
y = solvente / insolvente)

Regressão – problemas em que a variável resposta é um número real
(ex: y = vendas mensais
y = energia gerada
y = tempo de ciclo
y = retorno diário do BVSP)

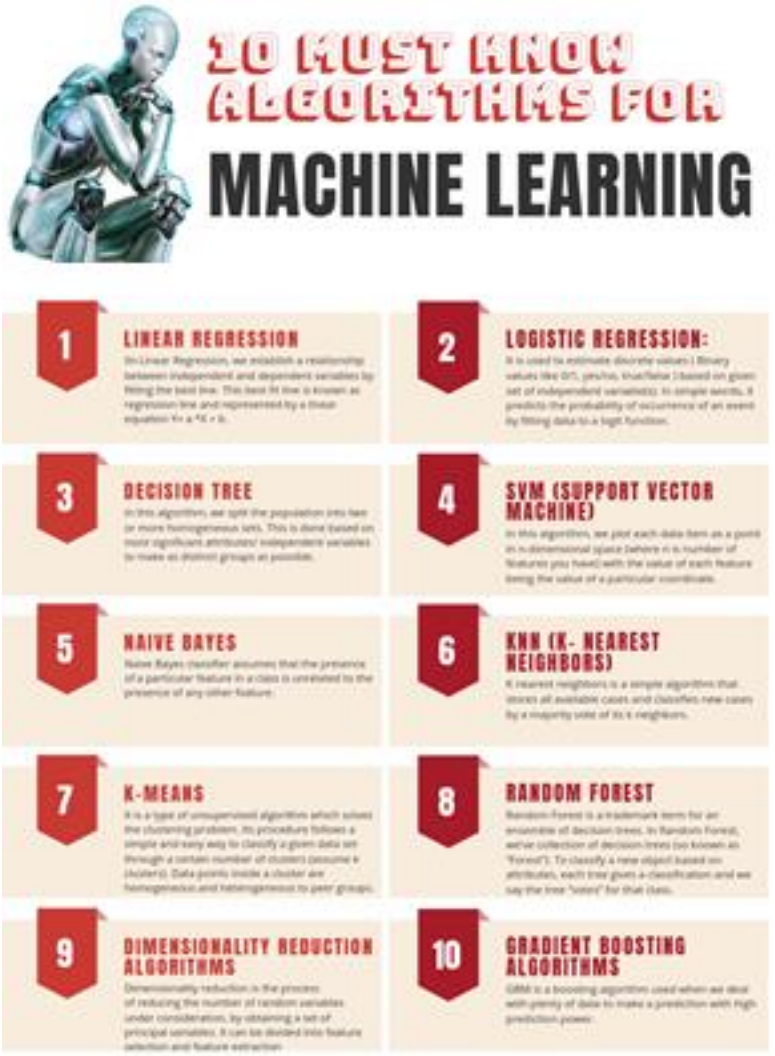
Fonte: https://vas3k.com/blog/machine_learning/

Machine Learning:



Machine Learning Algorithms in Python – You Must LEARN

Fonte: https://data-flair.training/blogs/machine-learning-algorithms-in-python/



Fonte: https://medium.com/@enochjoy/what-are-the-must-know-algorithms-for-machine-learning-3c6492757782

Programa do curso:

Seção	Conteúdo
1	Apresentação do curso. Introdução aos ambientes R e Rstudio. Construção de modelos de classificação. Caso 1: Classificador KNN.
2	Casos 2 e 3: Classificador Naïve Bayes e análise discriminante (linear e não-linear).
3	Caso 4: Classificadores baseados em regressão (Rlinear e Regressão logística).
4	Caso 5: Classificadores baseados em programação matemática e support vector machine (SVM).
5	Casos 6: Classificador CART e mistura de classificadores (Bagging, Boosting e Random Forest).
6	Caso 7: Avaliação de classificadores e práticas na construção de classificadores.
7	Construção de modelos de regressão. Casos 08 e 09: Regressão KNN e linear (simples e múltipla)
8	Caso 10: Regressão não-linear (GLM, polinomial, Splines e GAM)
9	Casos 11 e 12: Support vector regression, CART e modelos baseados em mistura (RG, GB)
10	Caso 13: Métodos prescritivos (otimização baseadas em modelos preditivos)

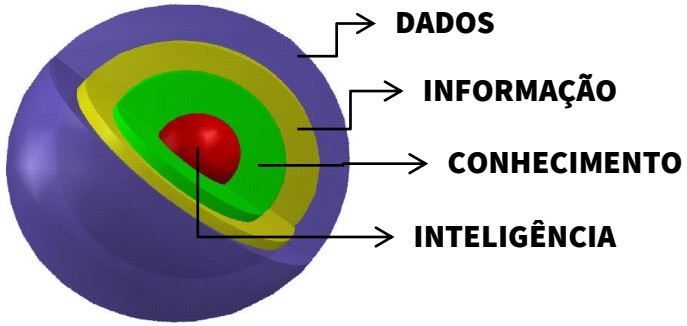
Bibliografia:

- Gareth, J., Witten, D., Hastie, T. e Tibshirani, R. An Introduction to Statistical Learning with applications in R. Springer, 2013.
- Alpaydin, E. Introduction to Machine Learning. MIT Press, 2004. Link: https://www.cmpe.boun.edu.tr/~ethem/i2ml/.
- Hastie, T., Tibshirani, R. e Friedman J, The Elements of Statistical Learning - Data Mining, Inference, and Prediction, 2nd edition, Springer, 2009.

Textos de apoio:

- Fayyad, U., Piatetsky-Shapiro, G. e Smyth, P. The KDD process for extracting useful knowledge from volumes of data. Communications of the ACM, vol. 39, no. 11, 1996.
- Tufekci, J. Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods. Electrical Power and Energy Systems, 60, 2014.

Introdução aos ambientes R e RStudio



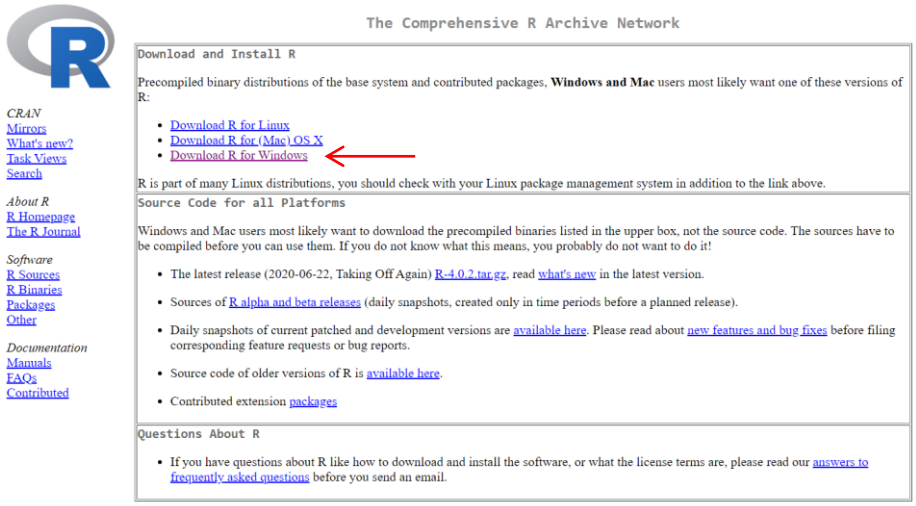
Rodrigo A. Scarpel

rodrigo@ita.br

www.ief.ita.br/~rodrigo

Introdução ao R e RStudio:

Site: https://cran.r-project.org/



The Comprehensive R Archive Network

CRAN
Mirrors
What's new?
Task Views
Search

About R
R Homepage
The R Journal

Software
R Sources
R Binaries
Packages
Other

Documentation
Manuals
FAQs
Contributed

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- Download R for Linux
- Download R for (Mac) OS X
- Download R for Windows

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

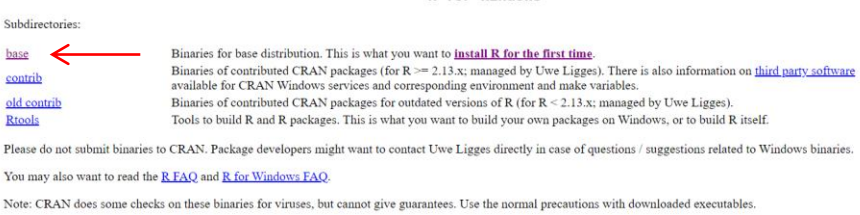
Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2020-06-22, Taking Off Again) R-4.0.2.tar.gz, read what's new in the latest version.
- Sources of R alpha and beta releases (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are available here. Please read about new features and bug fixes before filing corresponding feature requests or bug reports.
- Source code of older versions of R is available here.
- Contributed extension packages

Questions About R

- If you have questions about R like how to download and install the software, or what the license terms are, please read our answers to frequently asked questions before you send an email.



R for Windows

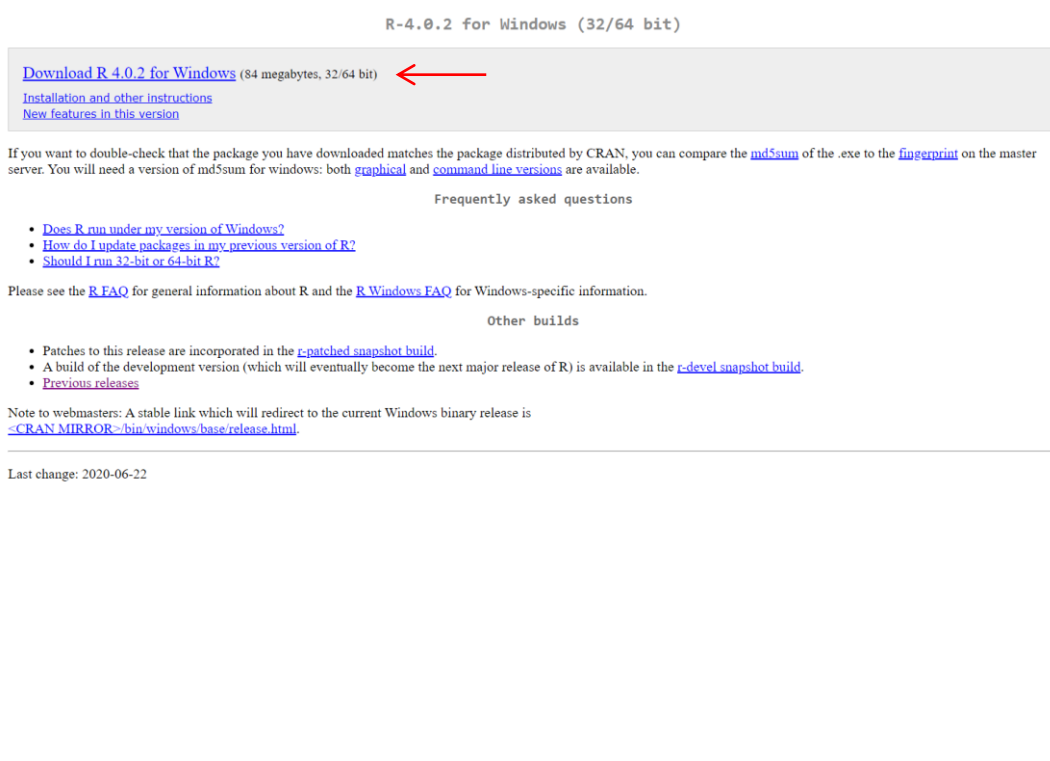
Subdirectories:

base — Binaries for base distribution. This is what you want to **install R for the first time**.
contrib — Binaries of contributed CRAN packages (for R >= 2.13.x; managed by Uwe Ligges). There is also information on third party software available for CRAN Windows services and corresponding environment and make variables.
old contrib — Binaries of contributed CRAN packages for outdated versions of R (for R < 2.13.x; managed by Uwe Ligges).
Rtools — Tools to build R and R packages. This is what you want to build your own packages on Windows, or to build R itself.

Please do not submit binaries to CRAN. Package developers might want to contact Uwe Ligges directly in case of questions / suggestions related to Windows binaries.

You may also want to read the R FAQ and R for Windows FAQ.

Note: CRAN does some checks on these binaries for viruses, but cannot give guarantees. Use the normal precautions with downloaded executables.



R-4.0.2 for Windows (32/64 bit)

Download R 4.0.2 for Windows (84 megabytes, 32/64 bit)
Installation and other instructions
New features in this version

If you want to double-check that the package you have downloaded matches the package distributed by CRAN, you can compare the md5sum of the .exe to the fingerprint on the master server. You will need a version of md5sum for windows: both graphical and command line versions are available.

Frequently asked questions

- Does R run under my version of Windows?
- How do I update packages in my previous version of R?
- Should I run 32-bit or 64-bit R?

Please see the R FAQ for general information about R and the R Windows FAQ for Windows-specific information.

Other builds

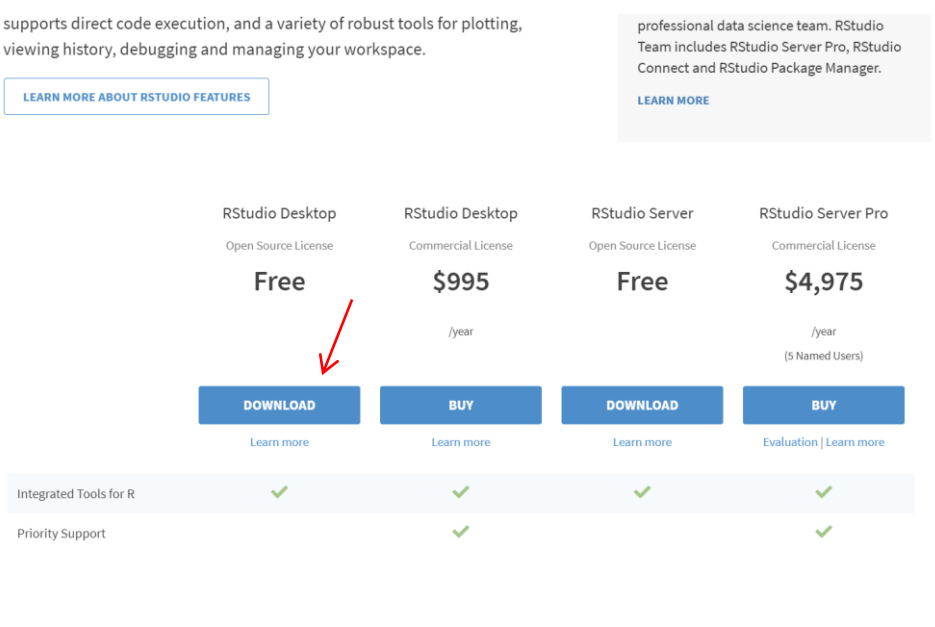
- Patches to this release are incorporated in the r-patched snapshot build.
- A build of the development version (which will eventually become the next major release of R) is available in the r-devel snapshot build.
- Previous releases

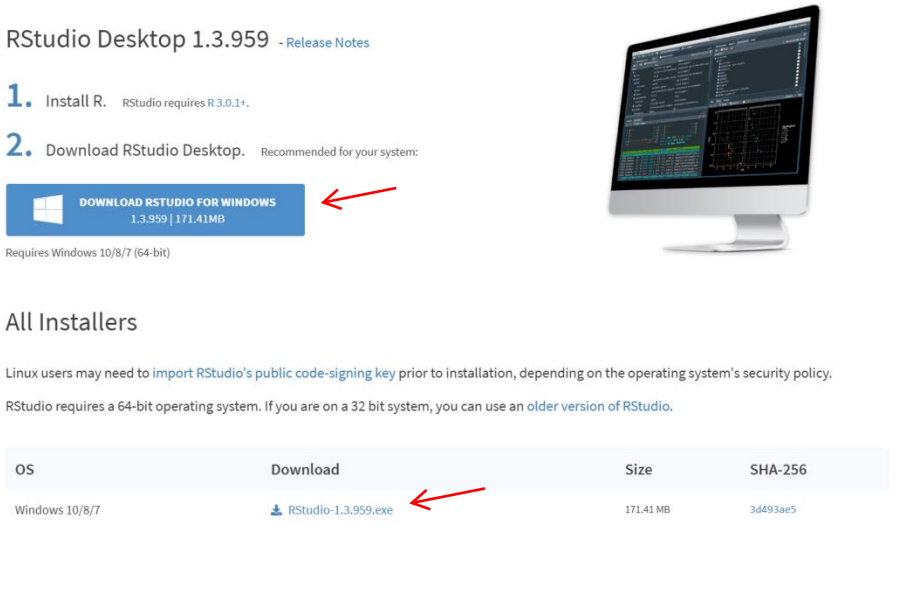
Note to webmasters: A stable link which will redirect to the current Windows binary release is <CRAN MIRROR>/bin/windows/base/release.html.

Last change: 2020-06-22

Introdução ao R e RStudio:

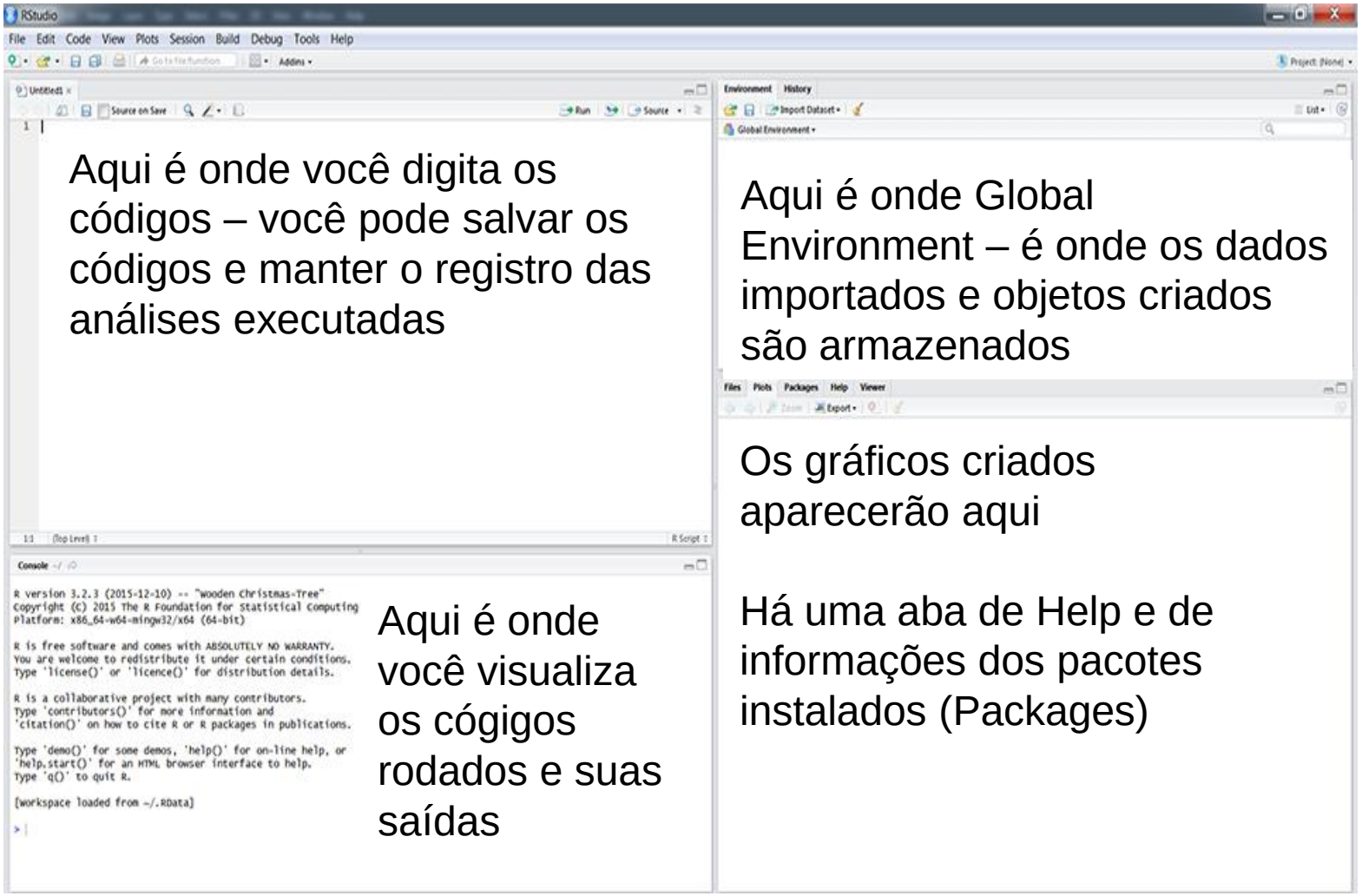
Site: https://rstudio.com/products/rstudio/download/





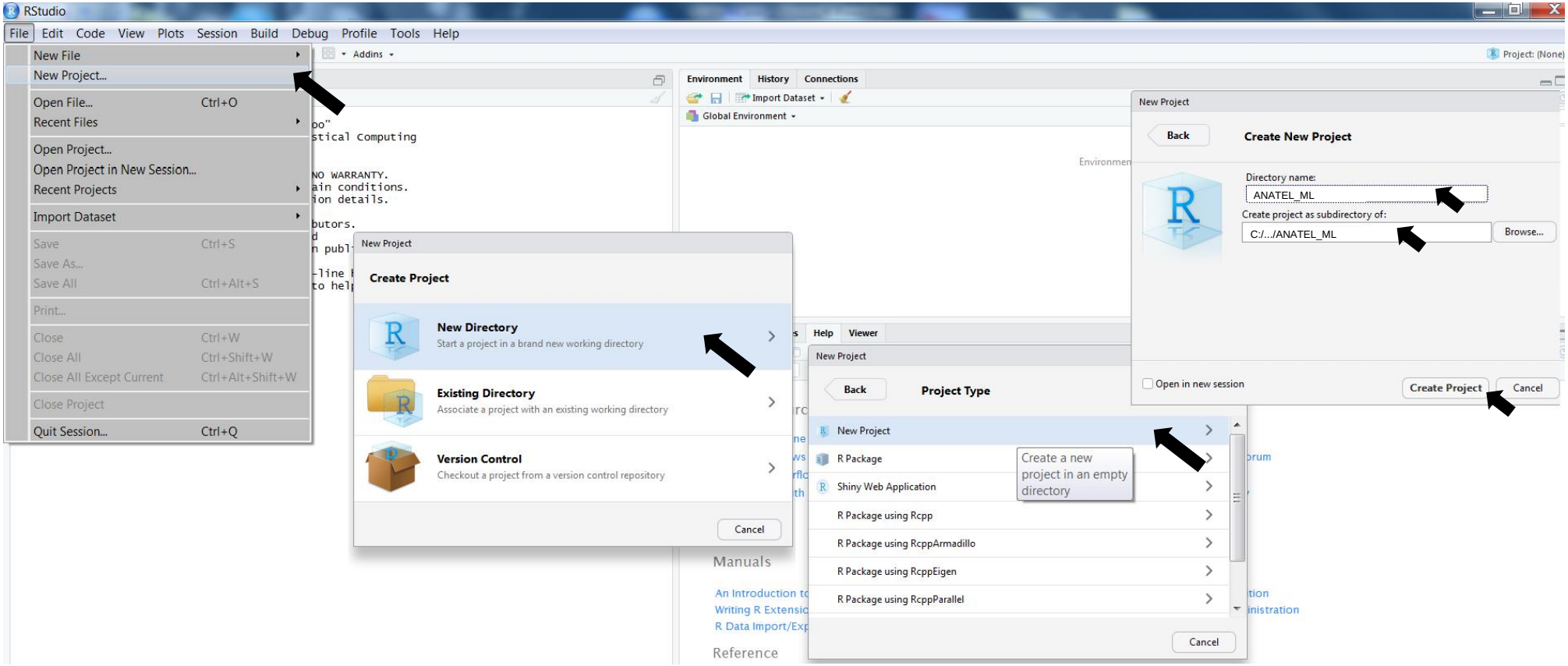
OBS: Só instale o RStudio depois do R já estar instalado

Introdução ao RStudio:



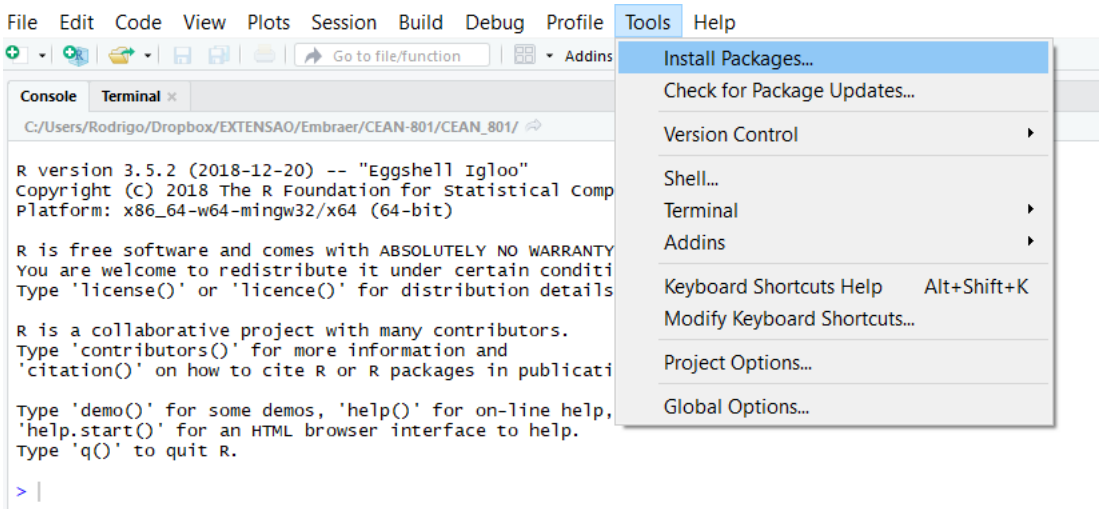
Introdução ao RStudio:

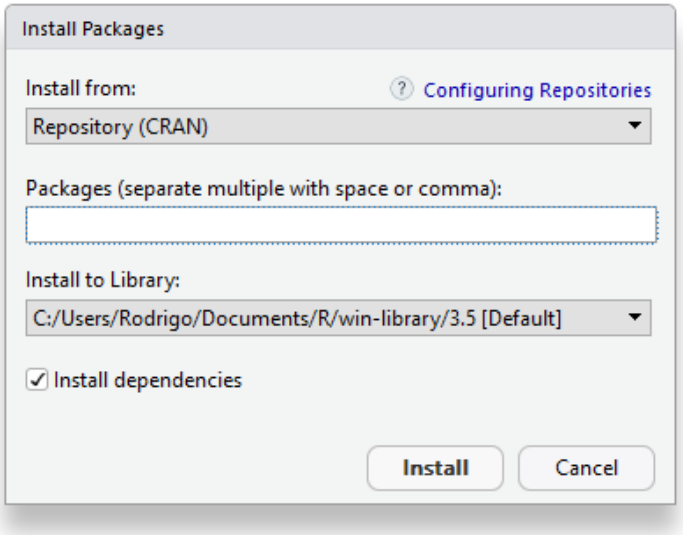
Criação de novos projetos:



Introdução ao RStudio:

Instalação dos pacotes (que serão utilizados):

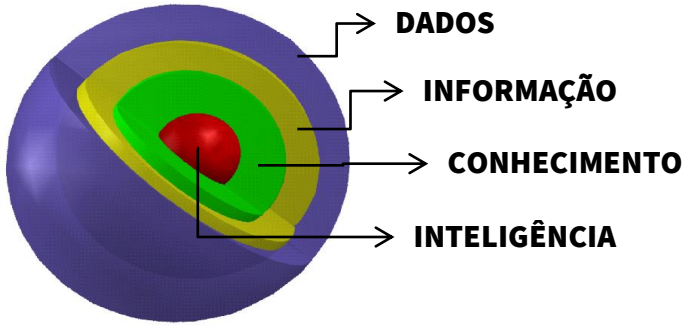




Pacotes:

• ggplot2		install.packages("ggplot2", dependencies=TRUE)
• dplyr		install.packages("dplyr", dependencies=TRUE)
• readxl	ou	install.packages("readxl", dependencies=TRUE)
• tidyr		install.packages(“tidyr", dependencies=TRUE)
• Lubridate		install.packages(“lubridate", dependencies=TRUE)
• ...		...

Construção de modelos de classificação



Rodrigo A. Scarpel

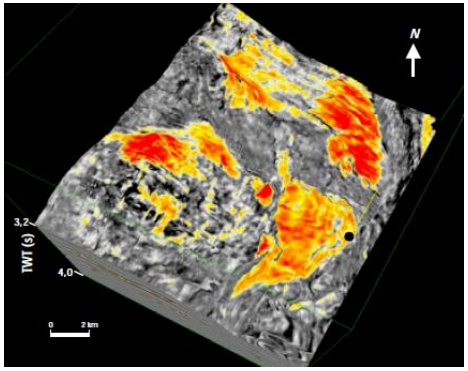
rodrigo@ita.br

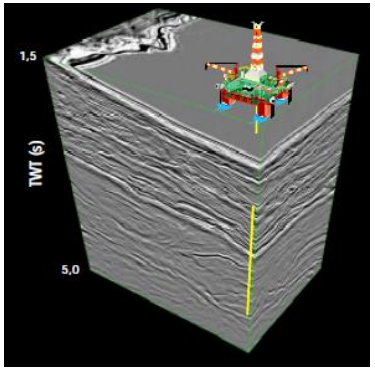
www.ief.ita.br/~rodrigo

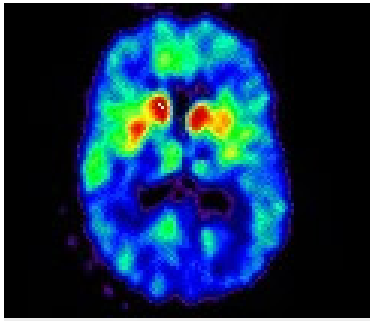
O problema de classificação:

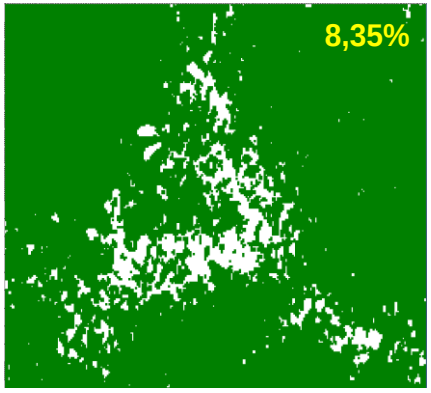
- Objetivo: classificar novas observações em categorias pré definidas.

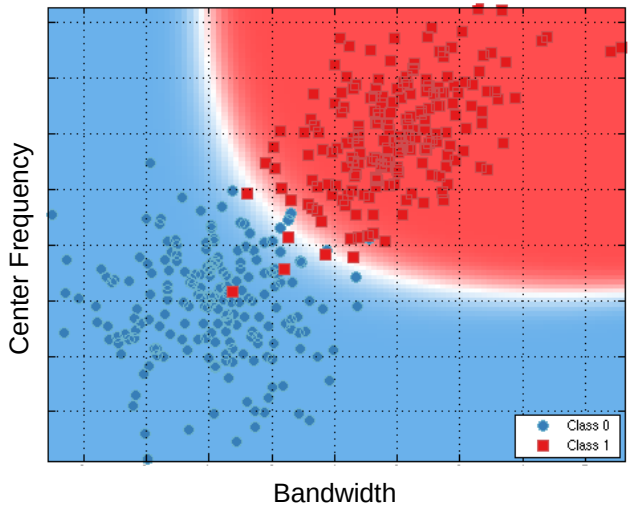
Exemplos:







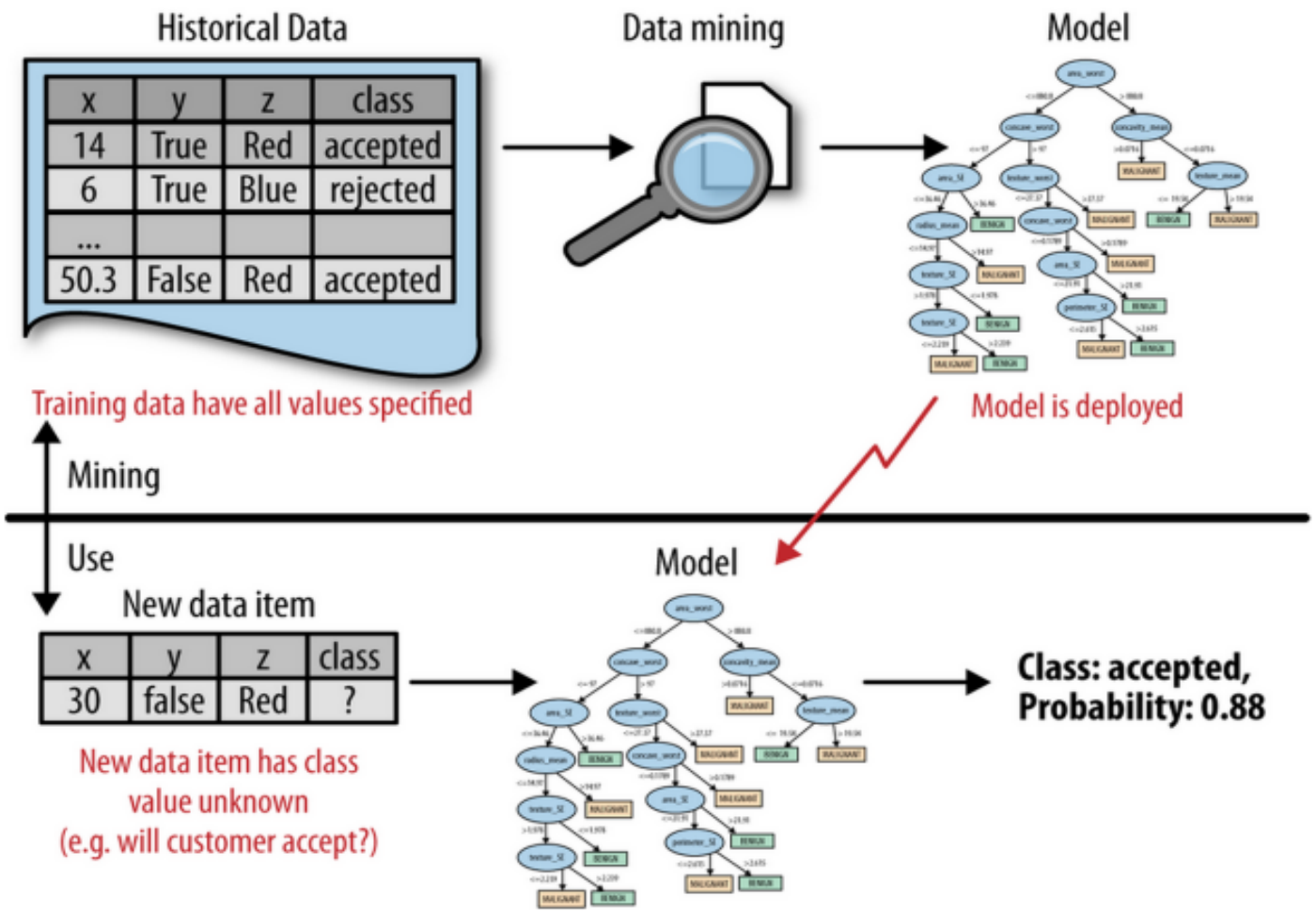




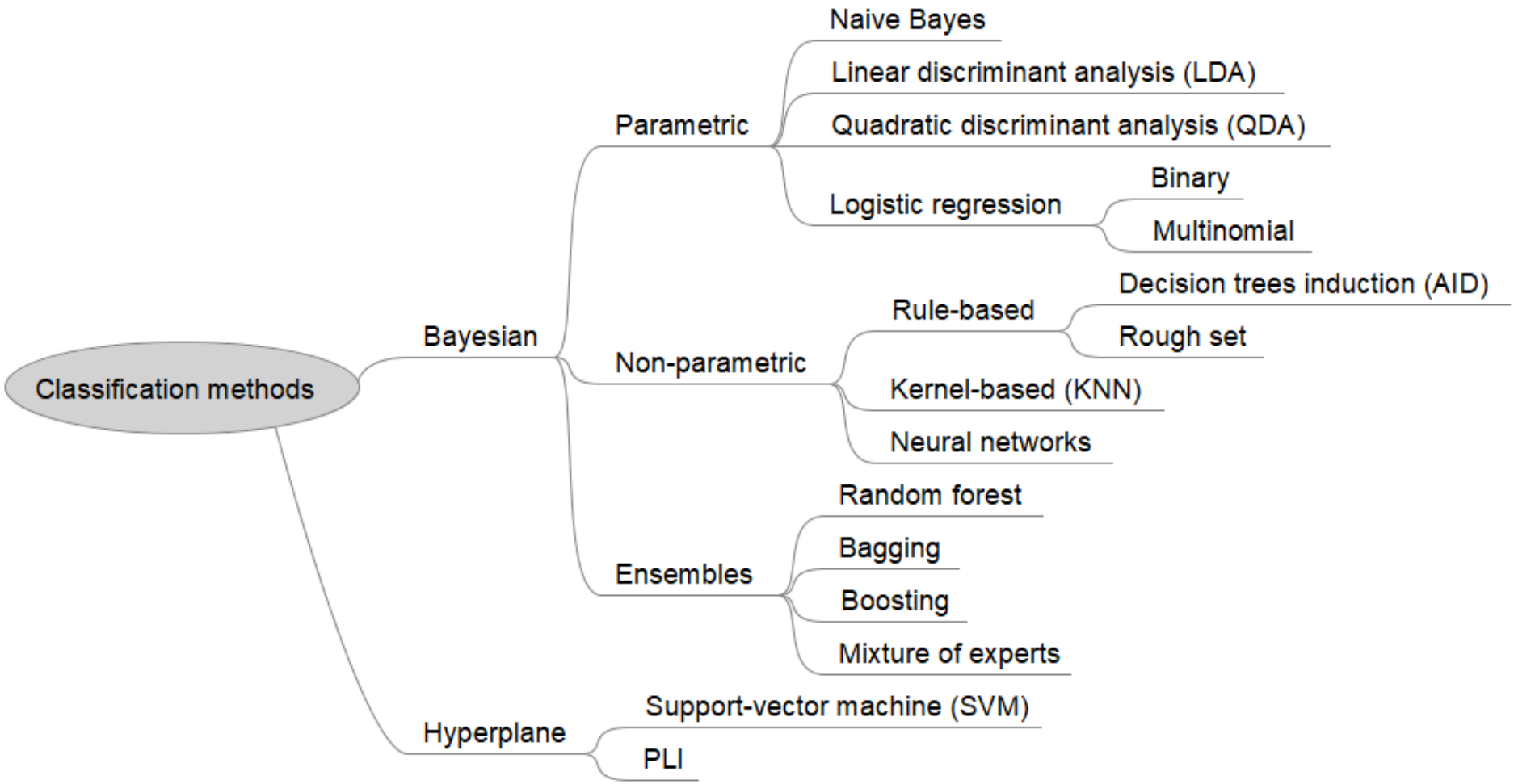
Métodos de classificação:

- São métodos utilizados para classificar novas em categorias pré definidas.

Exemplo:

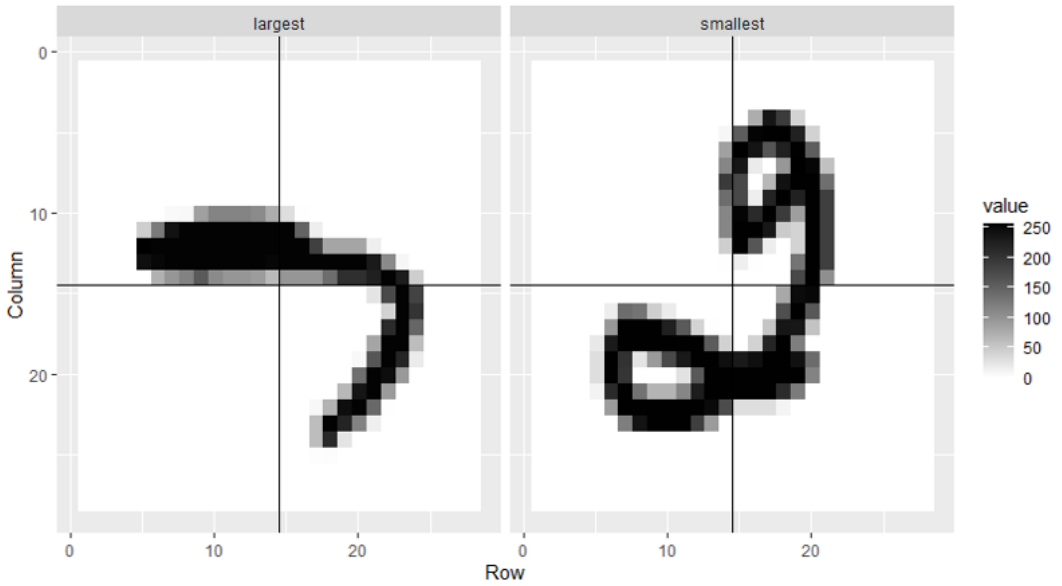


Métodos de classificação:



Métodos de classificação:

- **Casos 1 a 8: Classificação de dígitos (2 ou 7)**



Nova observação:

x_1 = 0.1253

x_2 = 0.2714

Classe ?

➢Atributos:

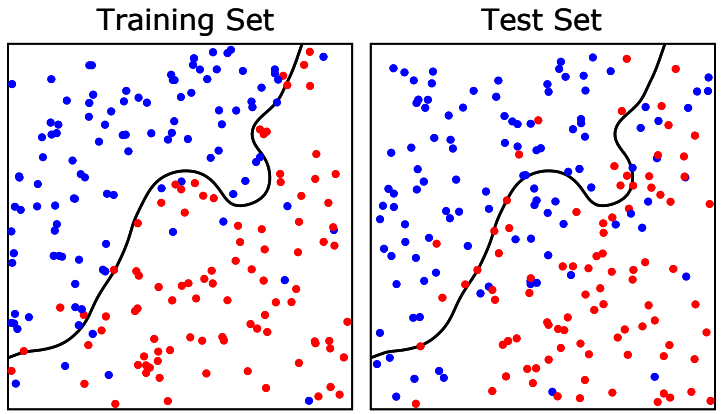
- x_1: proporção de pixels escuros no quadrante superior esquerdo
- x_2: proporção de pixels escuros no quadrante inferior direito

Construção de classificadores:

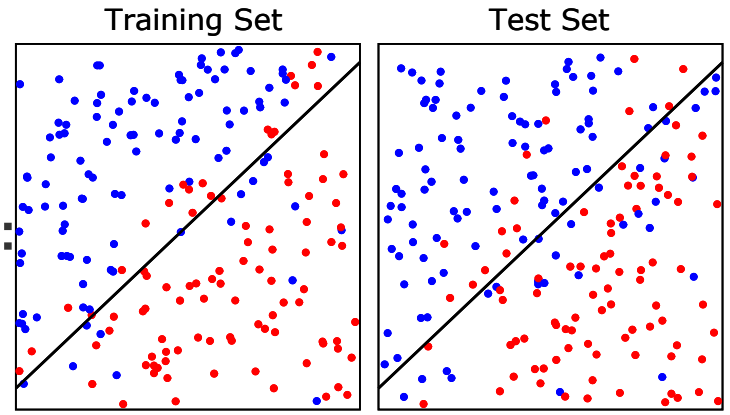
- Boas práticas na criação de modelos de classificação:

  Dividir os dados em bases de treinamento, validação e teste.

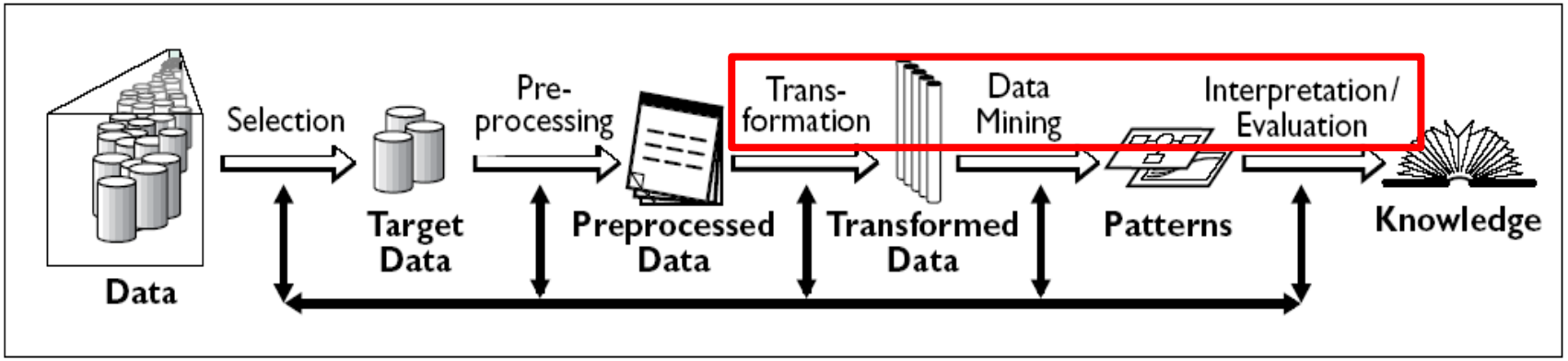
Overfitting:



Better Fitting:



Construção de classificadores:



DADOS DISPONÍVEIS PARA ANÁLISE

DADOS - TREINAMENTO

DADOS - TESTE → Acurácia: Teste (%)

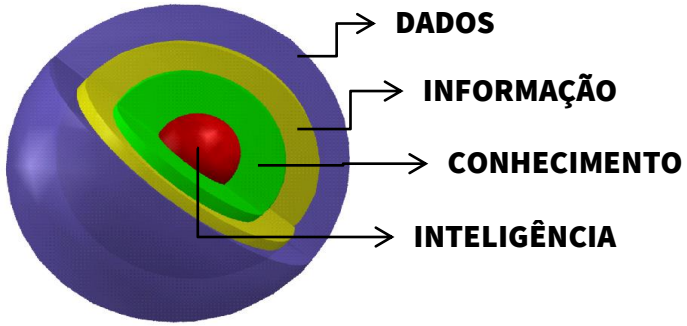
5–fold Cross-Validation (CV)

T : treino
V: validação

1:	T	T	T	T	V	→ Acurácia: T (T_1%) / V (V_1%)
2:	T	T	T	V	T	→ Acurácia: T (T_2%) / V (V_2%)
3:	T	T	V	T	T	→ Acurácia: T (T_3%) / V (V_3%)
4:	T	V	T	T	T	→ Acurácia: T (T_4%) / V (V_4%)
5:	V	T	T	T	T	→ Acurácia: T (T_5%) / V (V_5%)

Acurácia: média ($\bar{V}$%)

Classificador KNN

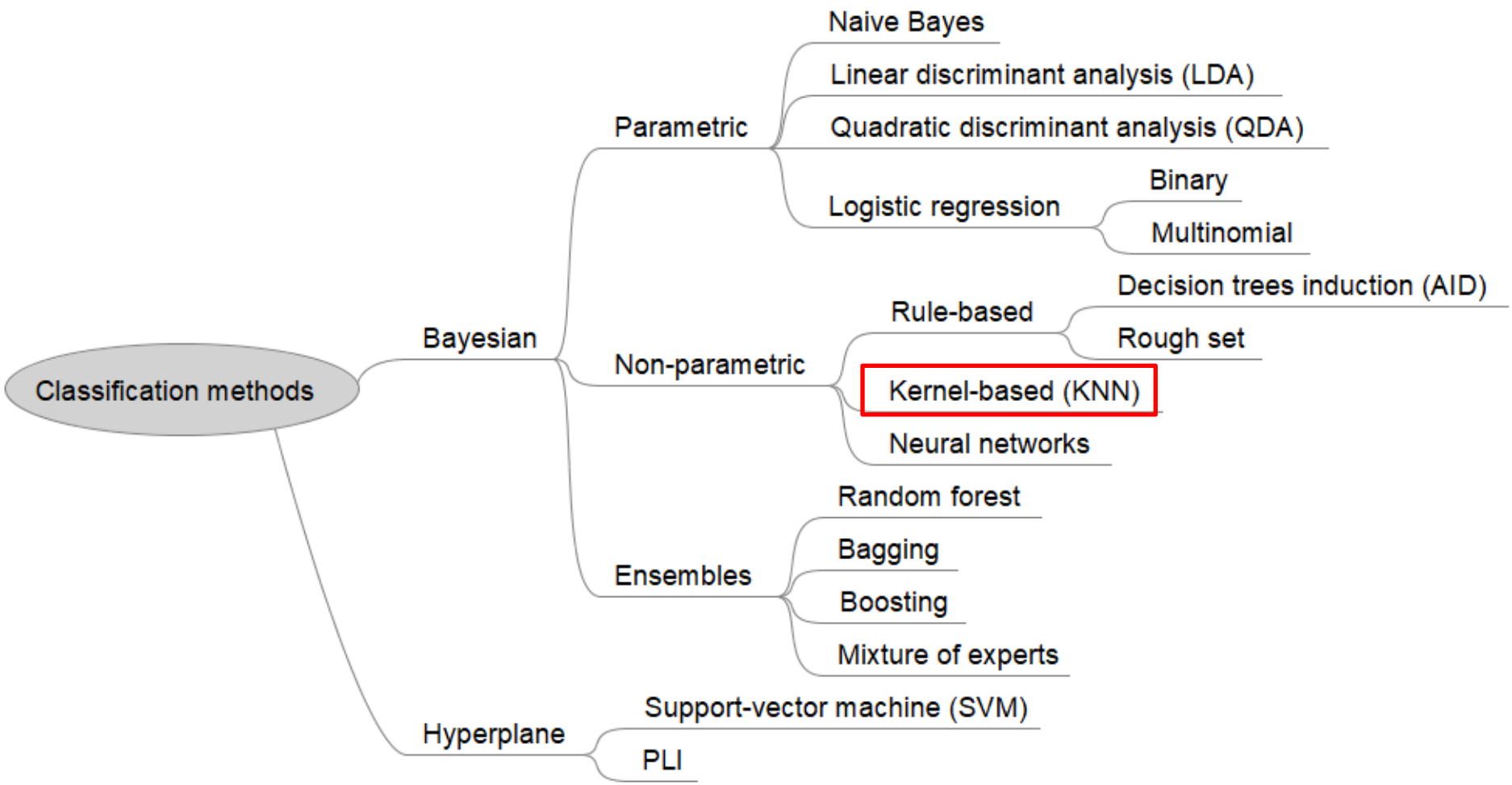


Rodrigo A. Scarpel

rodrigo@ita.br

www.ief.ita.br/~rodrigo

Métodos de classificação:



Classificadores Bayesianos:

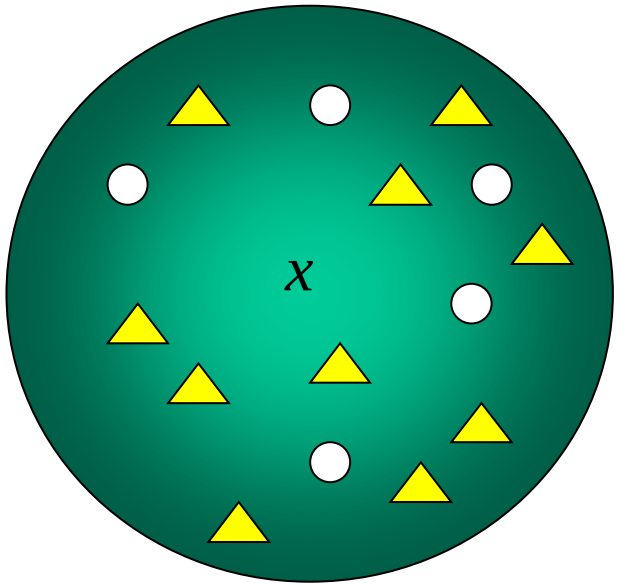
- Abordagem com os princípios utilizados pela maior parte dos métodos de classificação
- Também são conhecidos como classificadores de máxima verossimilhança
- Assume que todas as classes são conhecidas e que podem ser descritas de forma probabilística
- Sejam, no caso binário:
  - w_1 e w_2 as classes que queremos identificar
  - $p(w_1)$ e $p(w_2)$ as probabilidades das classes com $p(w_1) + p(w_2) = 1$
  - Classificação: se $p(w_1) > p(w_2)$ uma observação será classificada como sendo da classe w_1 e se $p(w_1) < p(w_2)$ a observação será classificada como sendo da classe w_2.
  - Erro do classificador: P(erro de classificação) = mínimo[$p(w_1)$, $p(w_2)$]

Caso 1: Classificador KNN

K-ésimo vizinho mais próximo (Discriminante não-paramétrico):

Para minimizar a probabilidade de erro de classificação de uma observação, toma-se os k vizinhos mais próximos (distância Euclideana) e atribui-se a observação à classe com a maior razão:

$$P(classe_j \mid x) = \frac{n_j}{k}$$

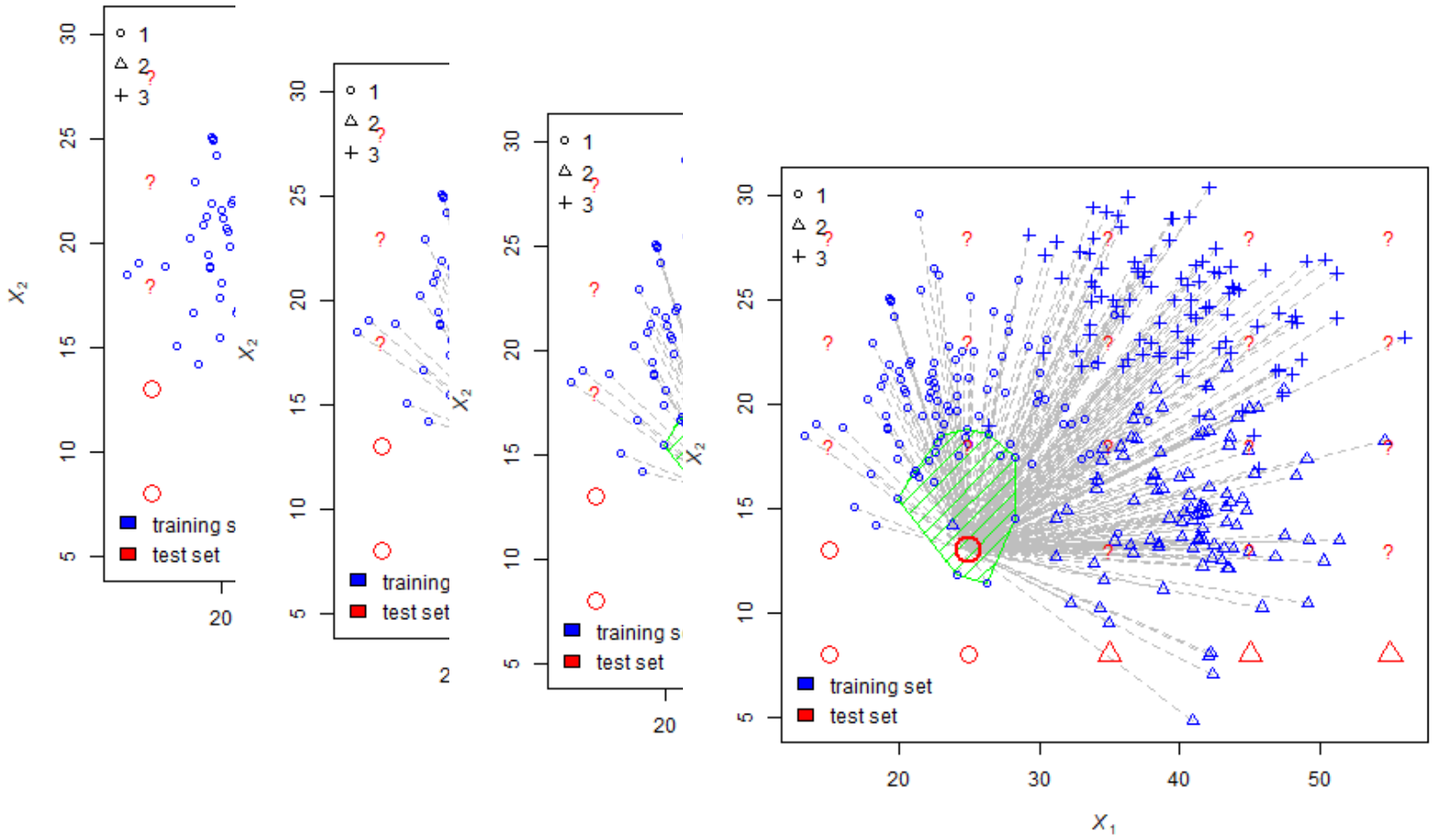


Classe	n_j	$P(c_j \mid x)$
○	5	0.33
△	10	0.66
Total (k)	15	1.00

$\therefore$ x = classe △

Caso 1: Classificador KNN

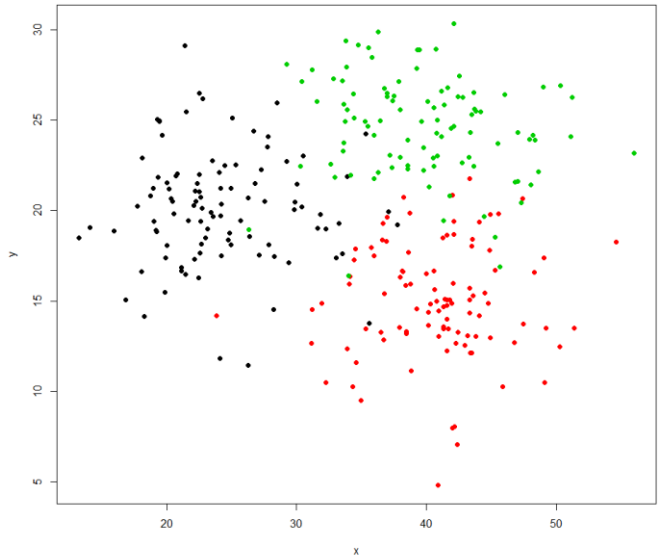
K-ésimo vizinho mais próximo: Exemplo

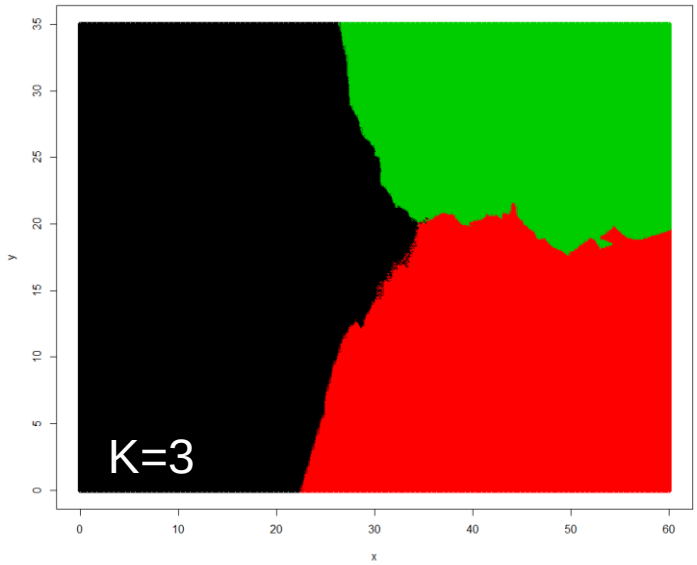


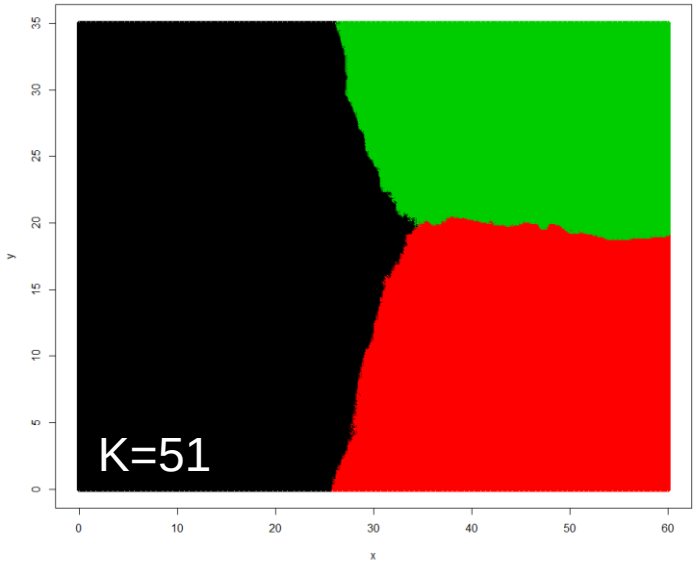
Link: http://www.ief.ita.br/~rodrigo/CURSOS/AO802/KNN/KNN.html

Caso 1: Classificador KNN

K-ésimo vizinho mais próximo: efeito do número de vizinhos

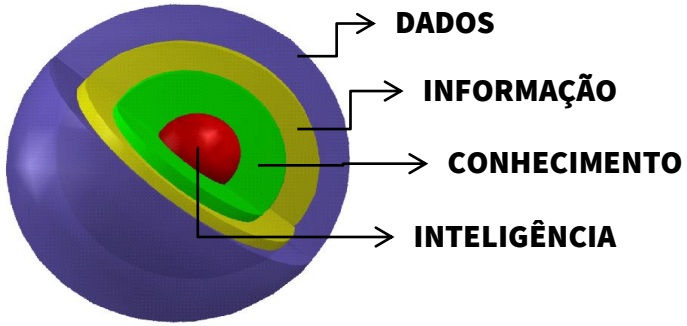






→ Escolha do K: validação cruzada (acurácia no conjunto de validação)

Classificador Naïve Bayes e Análise Discriminante

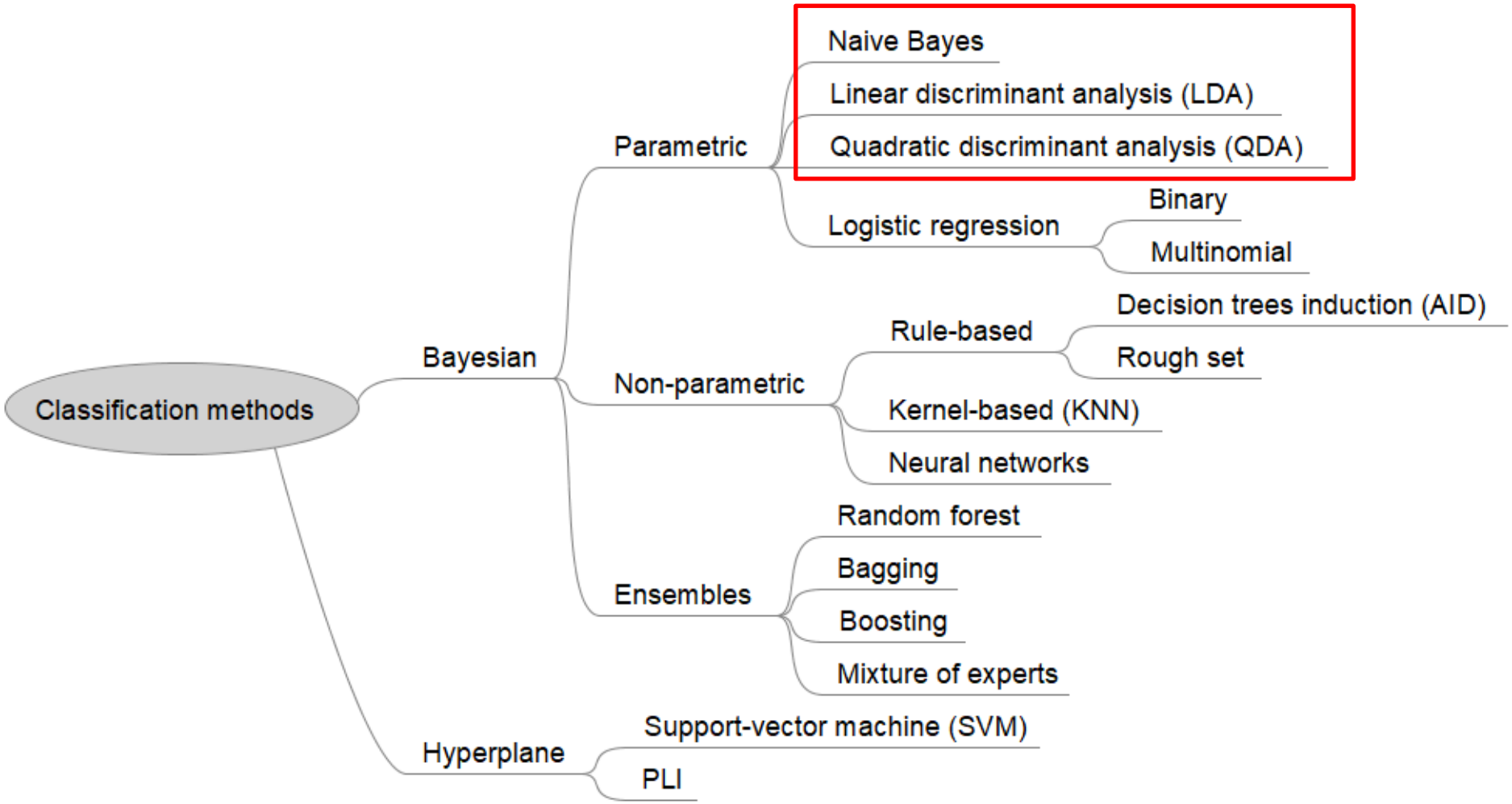


Rodrigo A. Scarpel

rodrigo@ita.br

www.ief.ita.br/~rodrigo

Métodos de classificação:



Caso 2: Classificador Naïve Bayes

Def: $p(x \backslash w_k)$ é a probabilidade de x, sabendo que é da classe w_k

Def: $p(x, w_k)$ é a probabilidade conjunta da informação x e da classe w_k e é obtida por $p(x \backslash w_k)$. $p(w_k)$

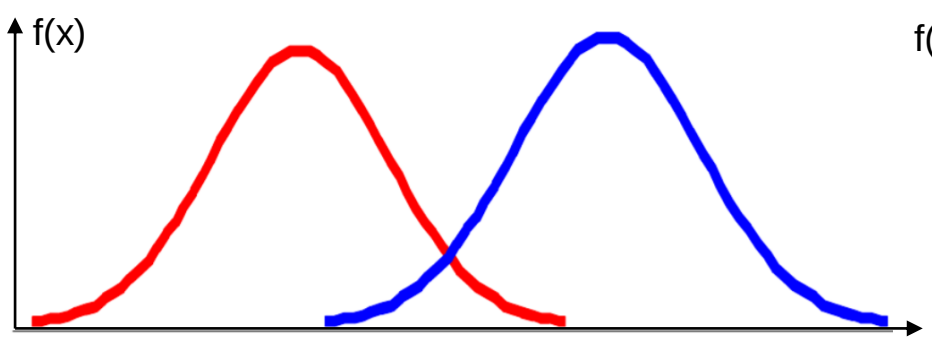
Portanto, Teorema da Probabilidade Total: $p(\mathbf{x}) = \sum_{k=1}^{K} p(\mathbf{x} \mid w_k) P(w_k)$

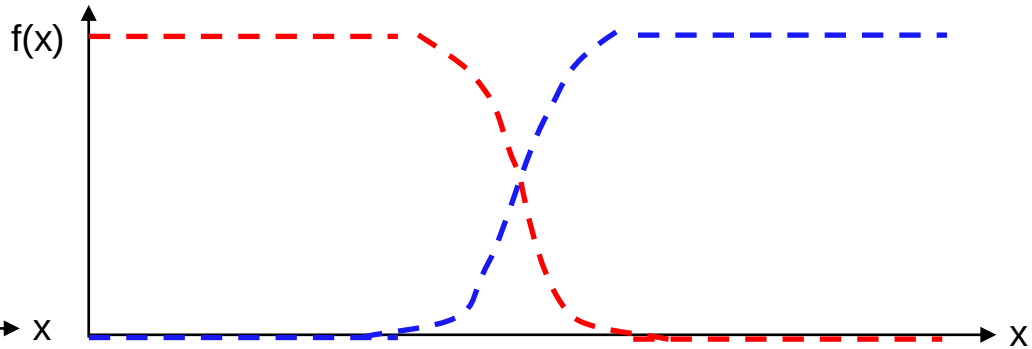
e pelo Teorema de Bayes:

prior *likelihood*

$$\textit{posterior} \rightarrow P(w \mid \mathbf{x}) = \frac{P(w)\, p(\mathbf{x} \mid w)}{p(\mathbf{x})}$$

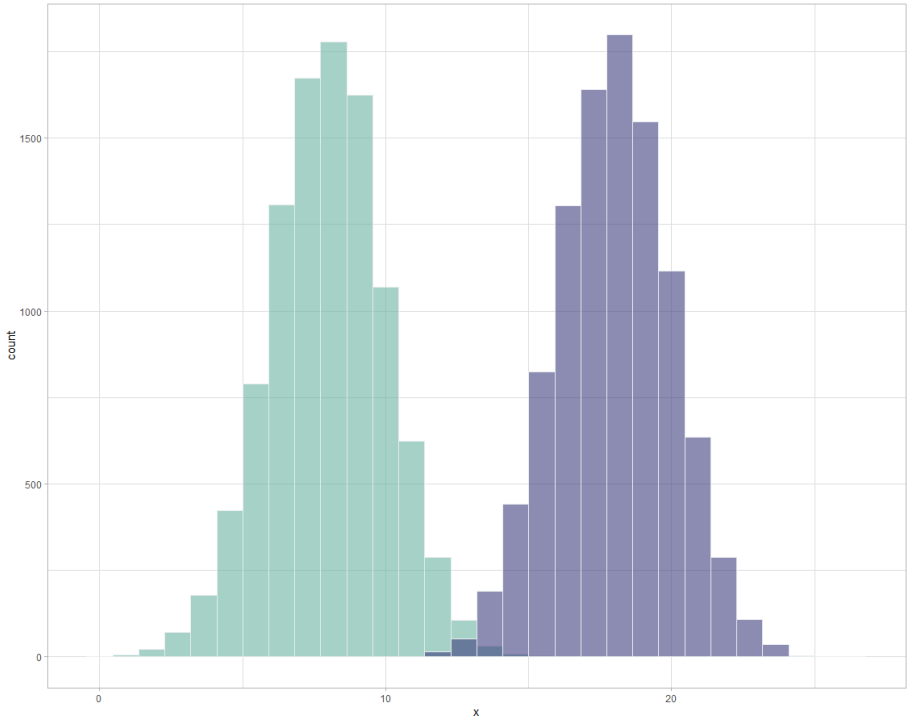
- Classificação (2 classes): se $p(w_1 \backslash x) > p(w_2 \backslash x)$ a observação será classificada como sendo da classe w_1 e da classe w_2, caso contrário.

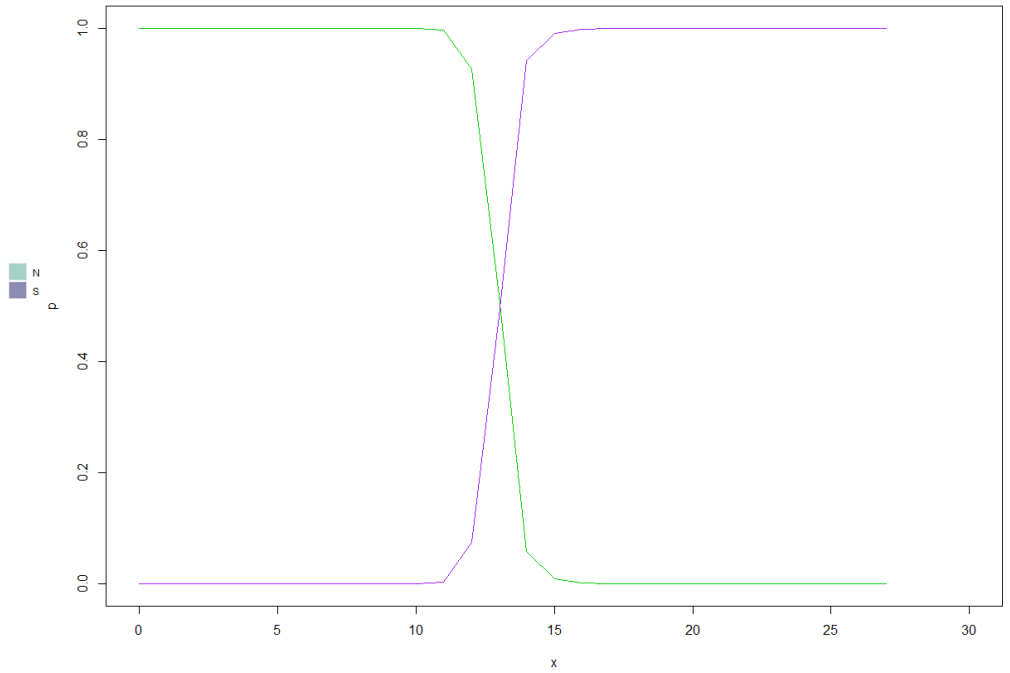




Caso 2: Classificador Naïve Bayes

- Situação 1: prioris iguais (p_1=50% e p_2 = 50%), variâncias iguais



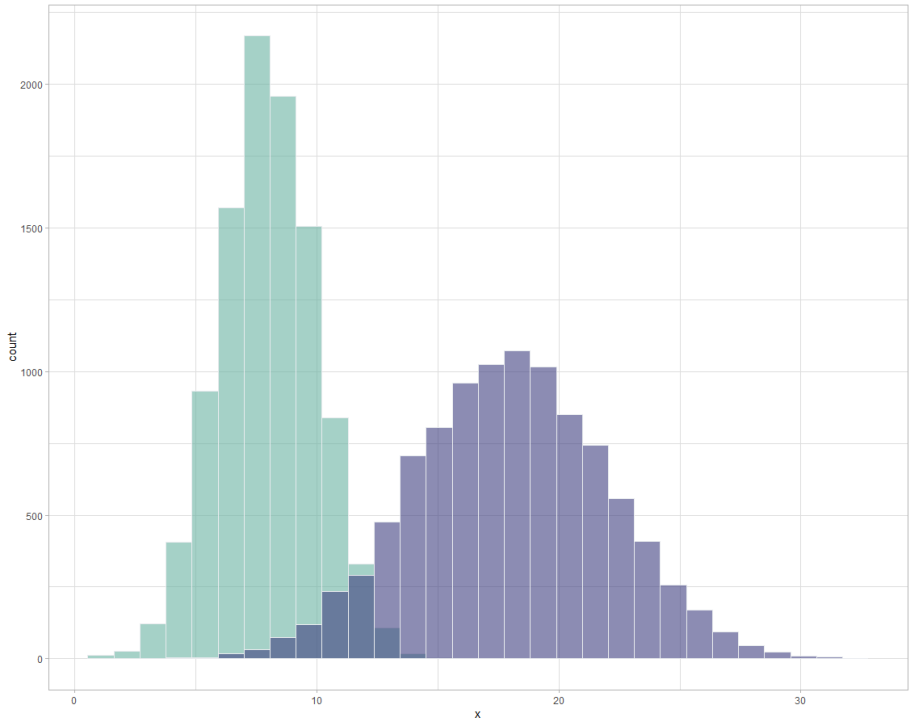


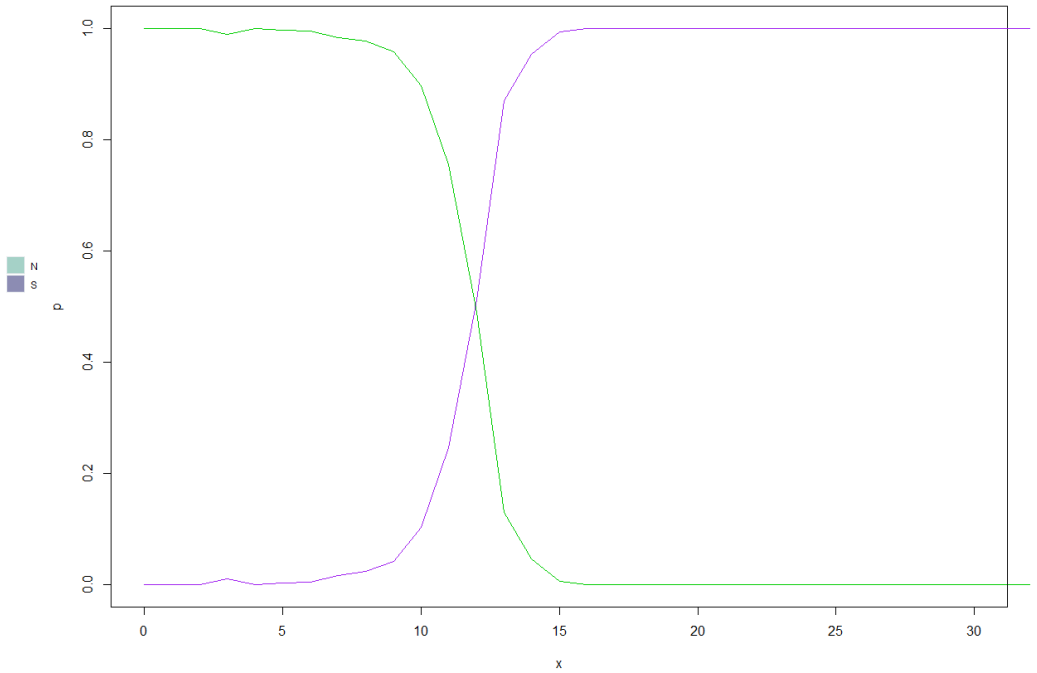
Não: média = 8 e desvio-padrão = 2

Sim: média = 18 e desvio-padrão = 2

Caso 2: Classificador Naïve Bayes

- Situação 2: prioris iguais (p_1=50% e p_2 = 50%), variâncias diferentes



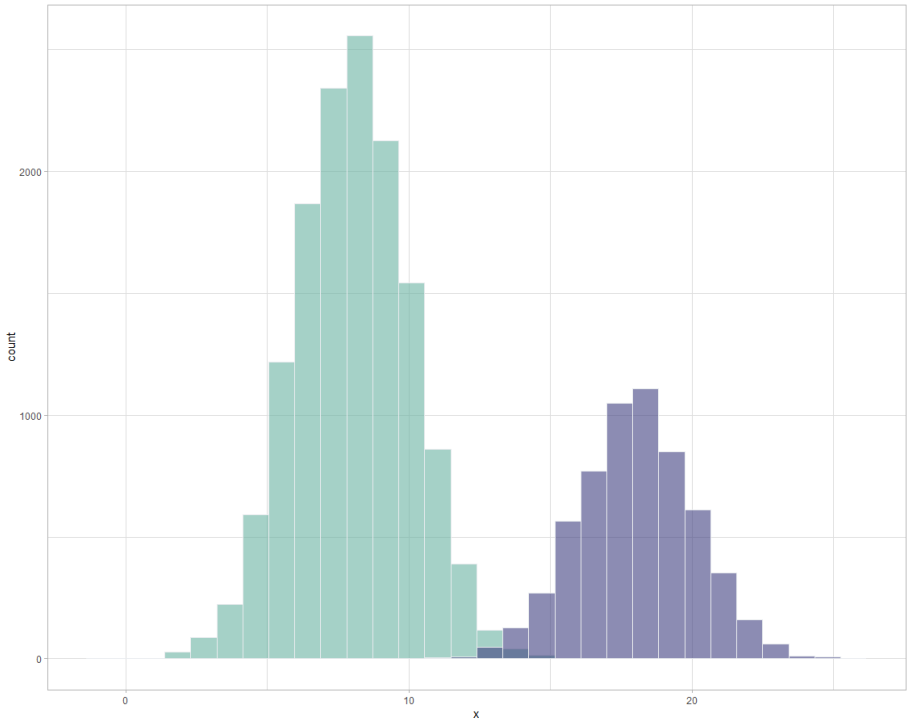


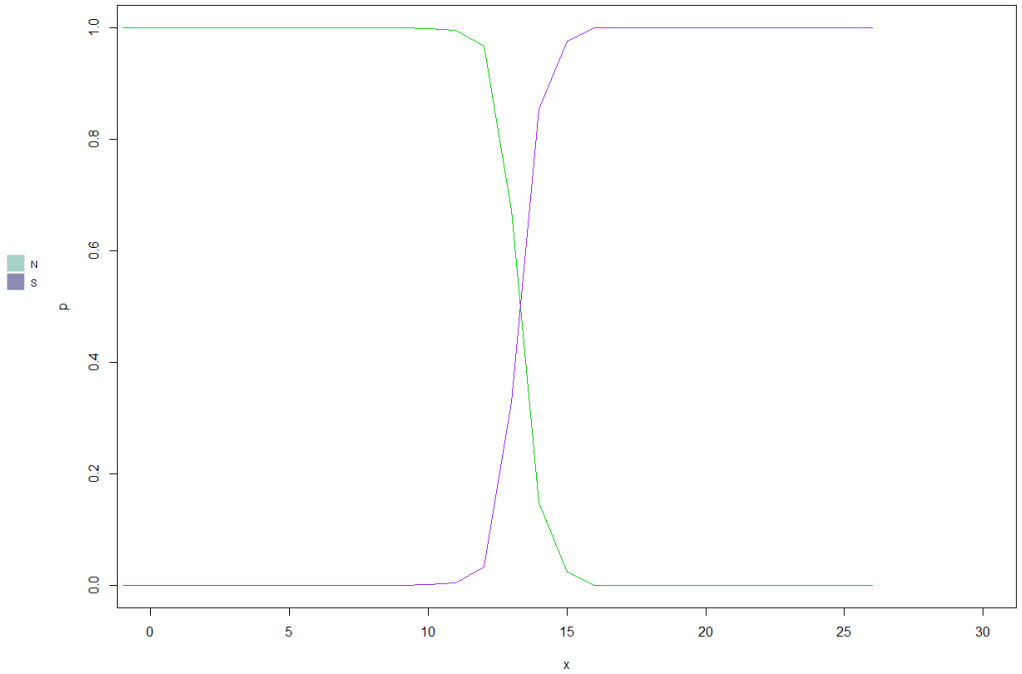
Não: média = 8 e desvio-padrão = 2

Sim: média = 18 e desvio-padrão = 4

Caso 2: Classificador Naïve Bayes

- Situação 2: prioris diferentes (p_1=70% e p_2 = 30%), variâncias iguais



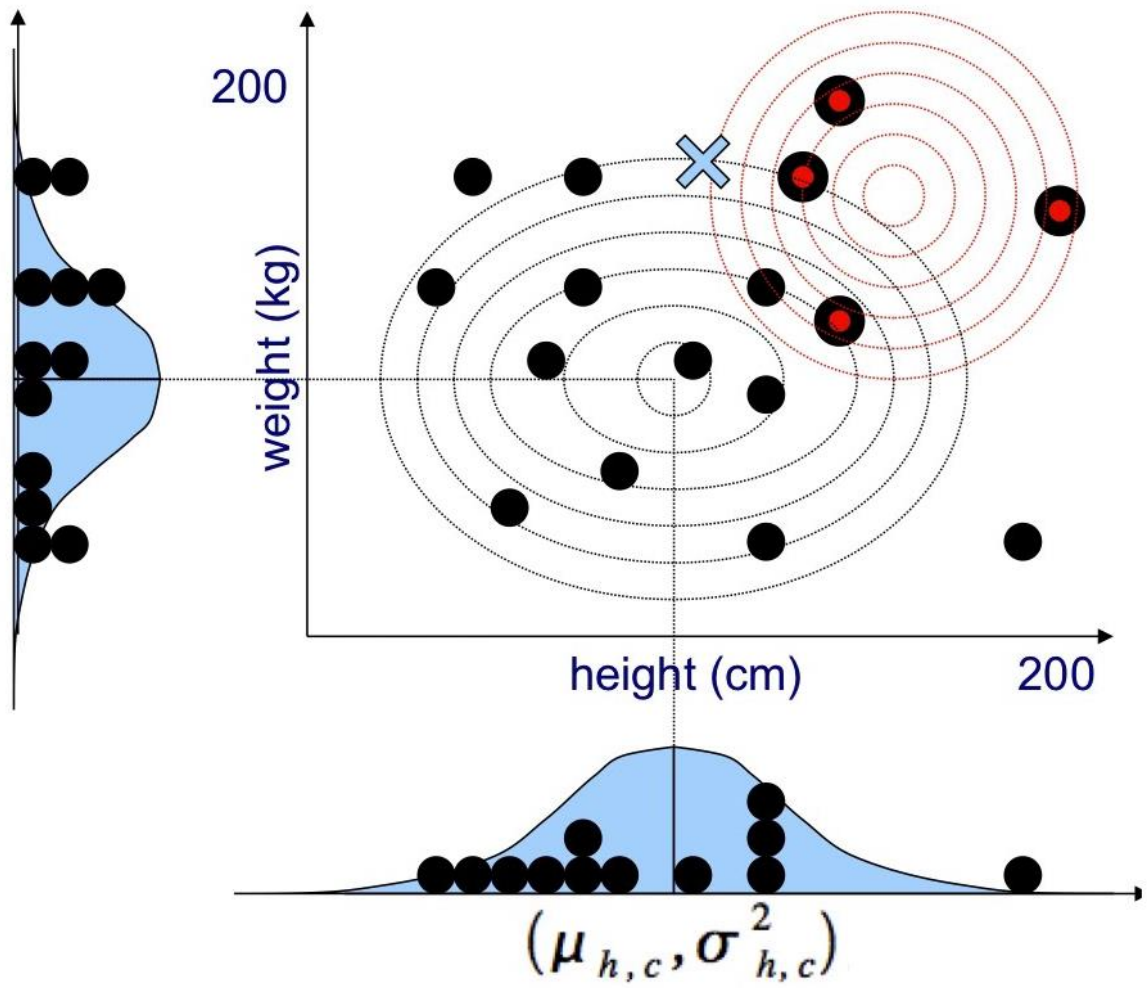


Não: média = 8 e desvio-padrão = 2

Sim: média = 18 e desvio-padrão = 2

Caso 2: Classificador Naïve Bayes

- Situação Multivariada:



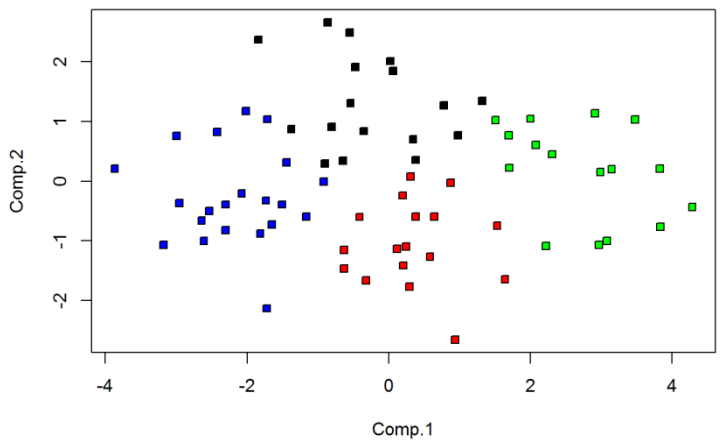
Para cada classe:

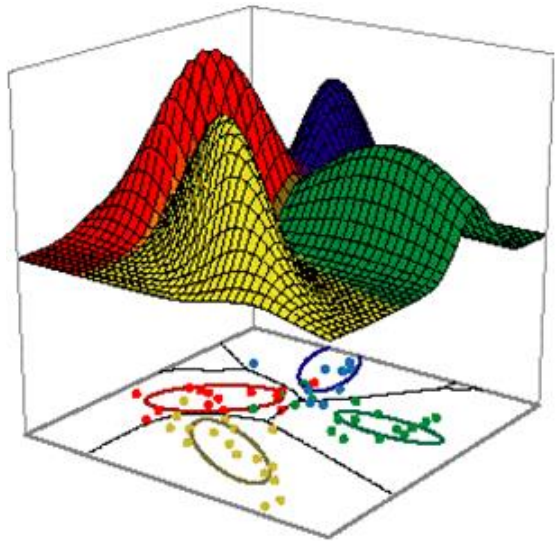
- Médias (das variáveis): μ_{x1}, ..., μ_{xn}
- Variâncias (das variáveis): σ^2_{x1}, ..., σ^2_{xn}

Caso 3: Análise Discriminante

- A análise discriminante (AD) é um método paramétrico de classificação.
- Por hipótese, acredita-se que as classes possuem distribuições de probabilidade distintas.
- A distribuição normal é comumente usada para representar as distribuições das classes.
- Opões: análise discriminante linear (LDA)

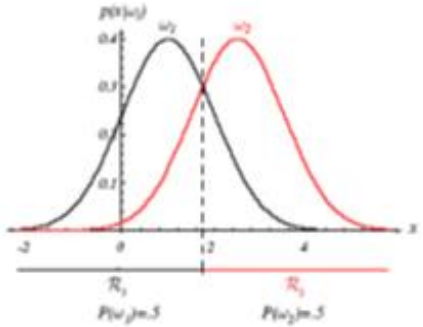
  análise discriminante quadrática (QDA)

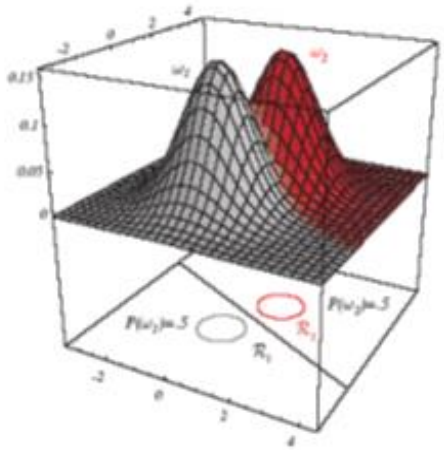


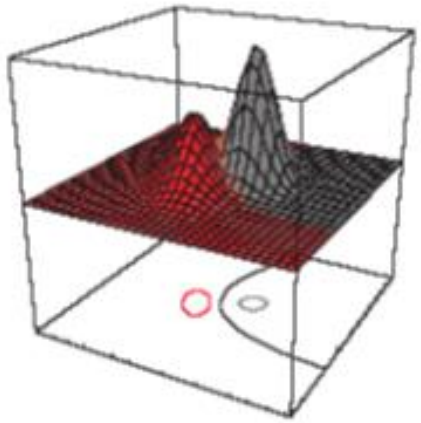


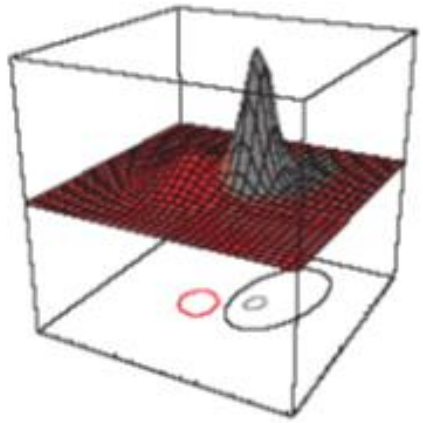
Caso 3: Análise Discriminante

Análise Discriminante Linear X Análise Discriminante Quadrática:





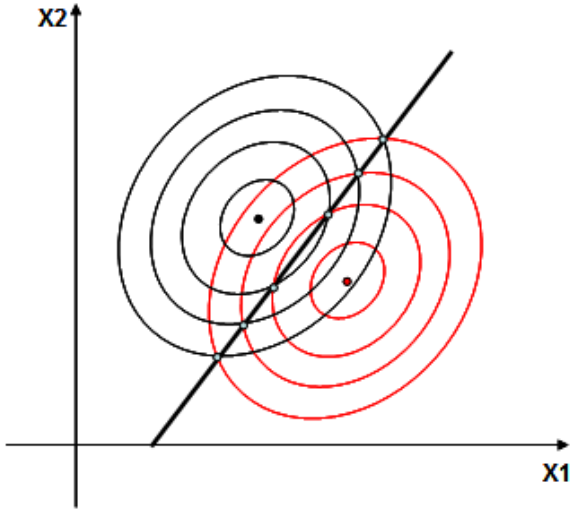




TREINAMENTO:

μ_i (para cada uma das classes)

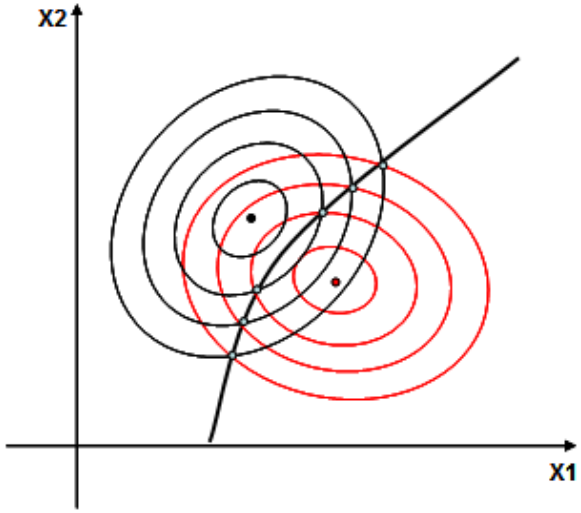
Σ (pooled)



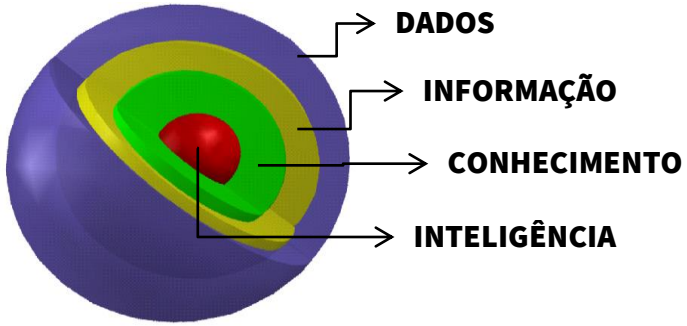
TREINAMENTO:

μ_i (para cada uma das classes)

Σ_i (para cada uma das classes)



Classificadores baseados em regressão

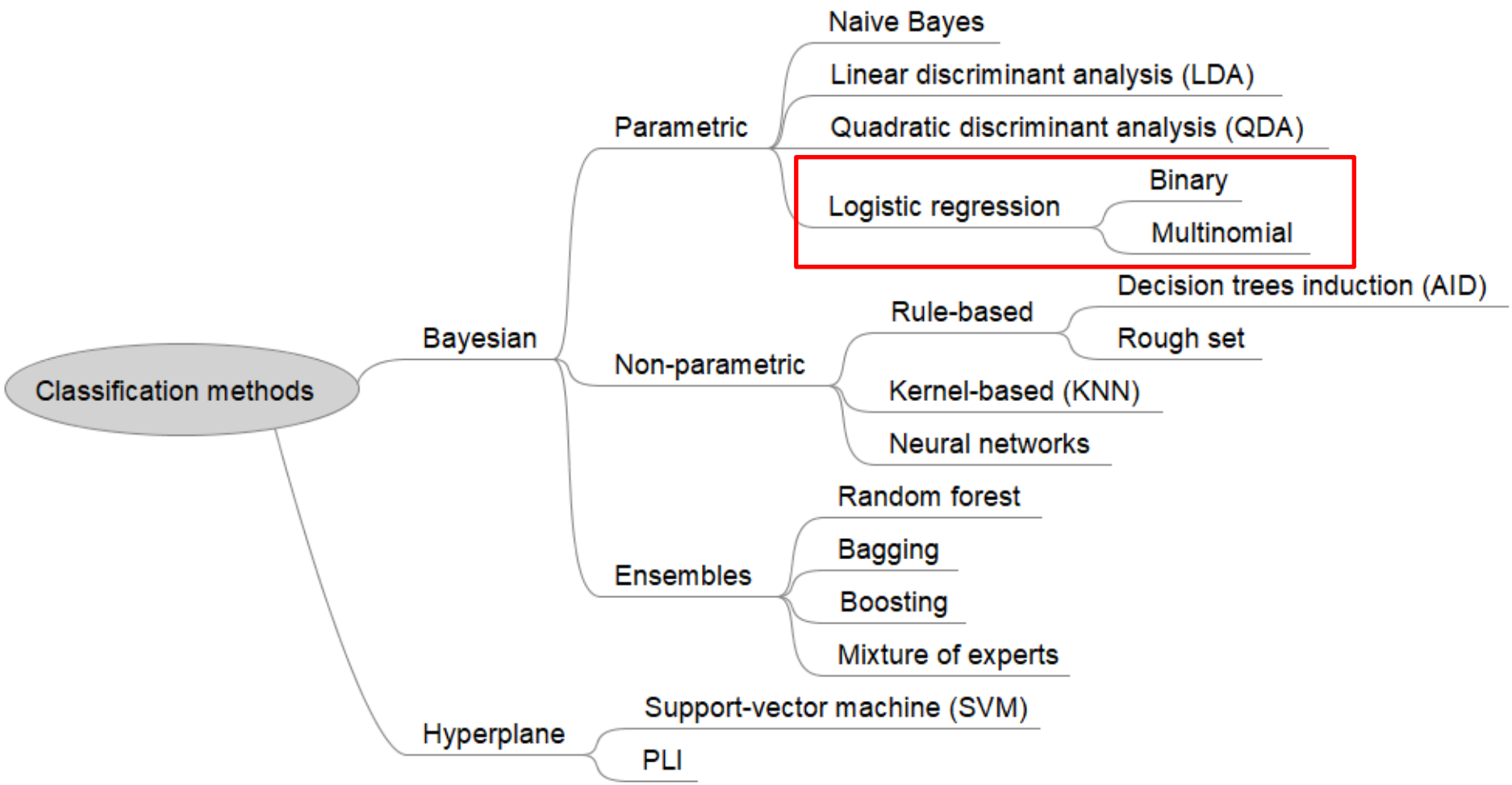


Rodrigo A. Scarpel

rodrigo@ita.br

www.ief.ita.br/~rodrigo

Métodos de classificação:



Classificadores por Regressão

- Métodos de regressão podem ser utilizados para a construção de modelo paramétricos ou não paramétricos de classificação.
  - Caso 1: objetiva-se a criação de uma função para estimar *p* (w_i | x,*θ*), em que x são as variáveis do modelo e *θ* são os parâmetros do modelo. Nesse caso

    Dados: $\mathcal{X} = \{x^t, w^t\}_{t=1}^N$ em que $x \in \Re$

    $$w_i^t = \begin{cases} 1 \text{ se } x^t \in C_i \\ 0 \text{ se } x^t \in C_j,\ j \neq i \end{cases}$$

    Classificação: uma observação é atribuída à classe C_i se $p(w_i | x,\theta) > p(w_j | x,\theta)$

  - Caso 2: objetiva-se a criação de uma função para estimar *f* (x | β), em que β são os parâmetros do modelo. Nesse caso

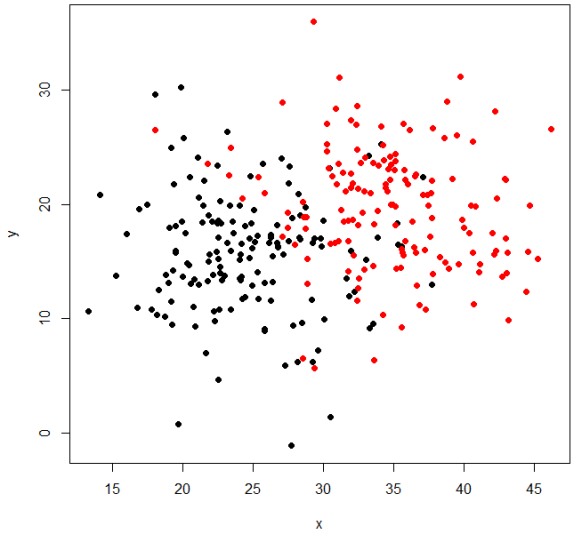
    Dados: $\mathcal{X} = \{x^t, w^t\}_{t=1}^N$ em que $x \in \Re$, $f_i(\mathbf{x}) = \beta_{i0} + \sum_{j=1}^{k} \beta_{ij} x_{ij}$

    $$w_i^t = \begin{cases} +1 \text{ se } x^t \in C_i \\ -1 \text{ se } x^t \in C_j,\ j \neq i \end{cases}$$

    Classificação: uma observação é atribuída à classe C_i se $f(x|\ \beta) > 0$

Classificadores por Regressão

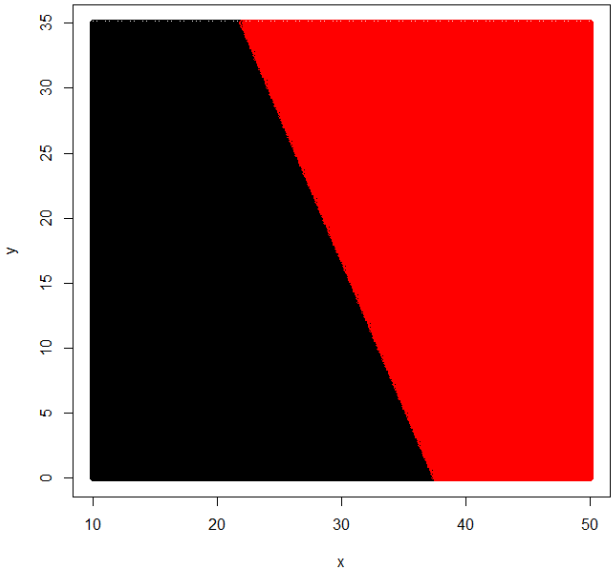
- Classificação por regressão linear:



Caso 1:

```
classe <- ifelse(dados$class == "0",0,1)
dados.rl = cbind(dados[,1:2],classe)
RLinear = lm(classe~x+y,data=dados.rl)
RLinear
CLASS.RLinear = as.factor(ifelse(predict(RLinear,z)<=0.5,0,1))
plot(z,col=unclass(CLASS.RLinear))
```

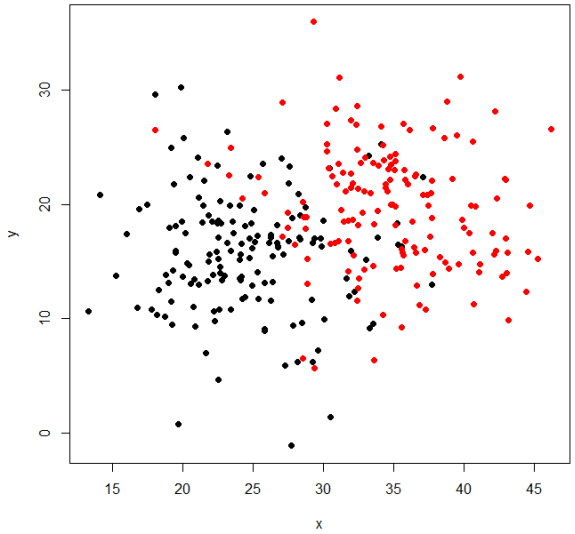
```
Call:
lm(formula = classe ~ x + y, data = dados.rl)

Coefficients:
(Intercept)            x            y
   -1.28010      0.04749      0.02109
```



$$\rightarrow p(\text{Vermelho} \mid \mathbf{x}, \beta) = -1{,}28 + 0{,}047x + 0{,}021y$$

Classificadores por Regressão

- Classificação por regressão linear:

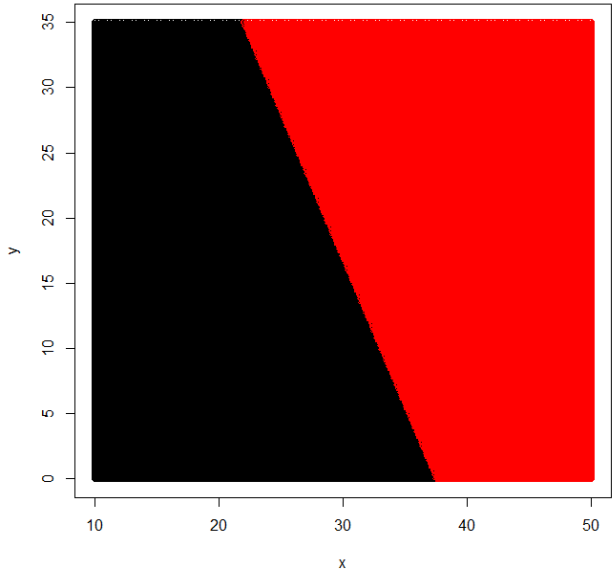


Caso 2:

```
classe <- ifelse(dados$class == "0",-1,1)
dados.rl2 = cbind(dados[,1:2],classe)
RLinear2 = lm(classe~x+y,data=dados.rl2)
RLinear2
CLASS.RLinear2 = as.factor(ifelse(predict(RLinear2,z)<=0,-1,1))
plot(z,col=unclass(CLASS.RLinear2))
```

```
Call:
lm(formula = classe ~ x + y, data = dados.rl2)

Coefficients:
(Intercept)            x            y
   -3.56020      0.09499      0.04219
```

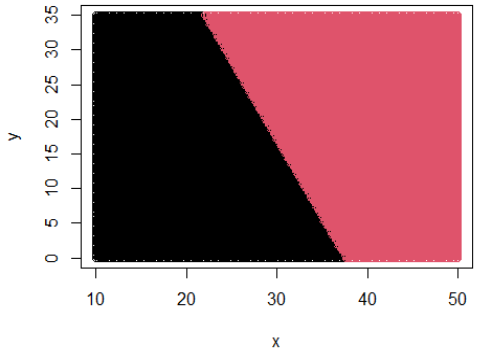
$$\rightarrow f(x, y \mid \beta) = -3{,}56 + 0{,}095x + 0{,}042y$$



Classificadores por Regressão

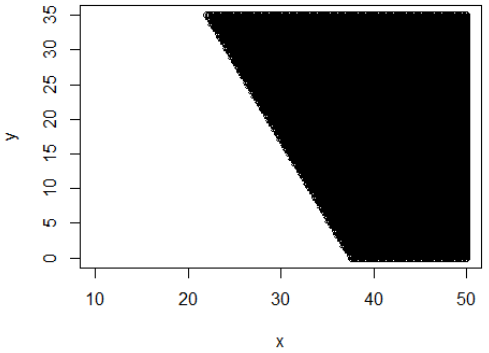
- Equivalência entre a classificação por regressão linear e LDA:

```
# LDA:
fit.LDA = train(classe ~ x + y,
                method ='lda',
                data = dados)
Prev.LDA <- predict(fit.LDA,newdata = z)
plot(z,col=unclass(Prev.LDA))
```




```
# z:
x = seq(10,50,0.2)
y = seq(0,35,0.2)
z = data.frame(expand.grid
                x,y))
```

```
# Regressão linear:
classe <- ifelse(dados$classe=='1',1,0)
dados.rl <- data.frame(x=dados$x,y=dados$y,classe=classe)
fit.RL = train(classe ~ x + y, method ='lm',data = dados.rl)
Prev.RL <- predict(fit.RL,newdata = z)
Prev.RL.CLASS <- ifelse(Prev.RL >= 0.5,1,0)
plot(z,col=unclass(Prev.RL.CLASS))
```




```
# Comparação das classificações:
table(Prev.LDA,Prev.RL.CLASS)
```

Prev.LDA	Prev.RL.CLASS = 0	Prev.RL.CLASS = 1
0	17428	0
1	0	17948

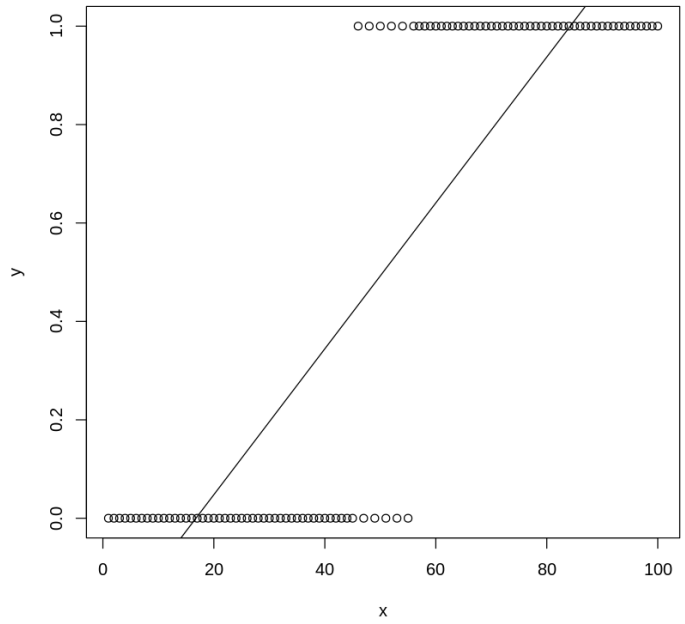
Caso 4: Classificador por Regressão Logística

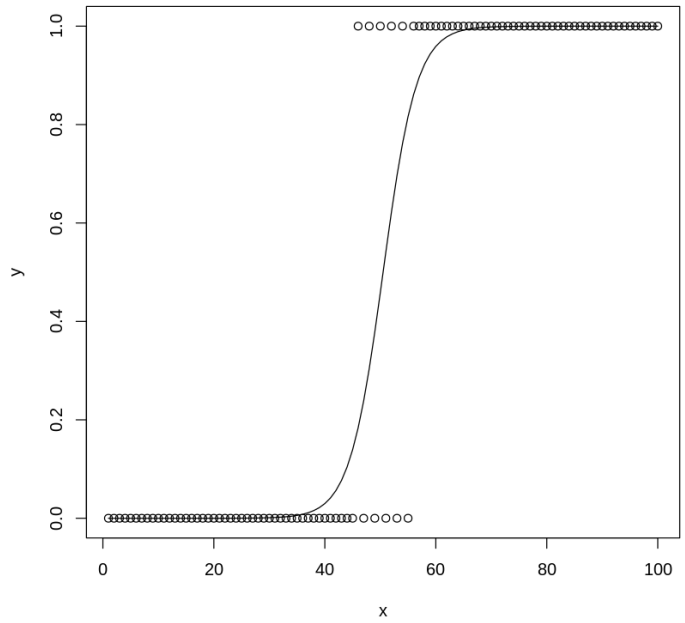
Regressão Linear

$$y_i = \alpha + \beta x_i + \varepsilon_i$$

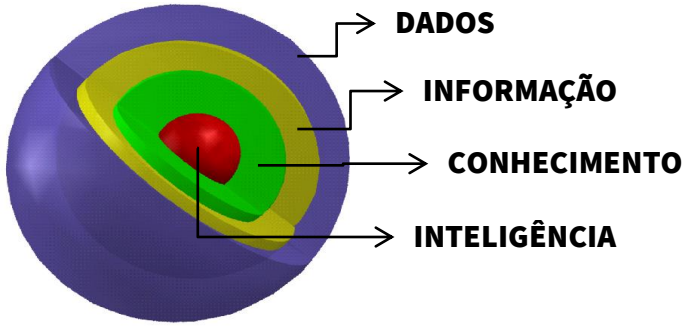
Regressão Logística

$$y_i = \frac{1}{1 + e^{-(\alpha + \beta x_i + \varepsilon_i)}}$$





Classificadores baseados em programação matemática

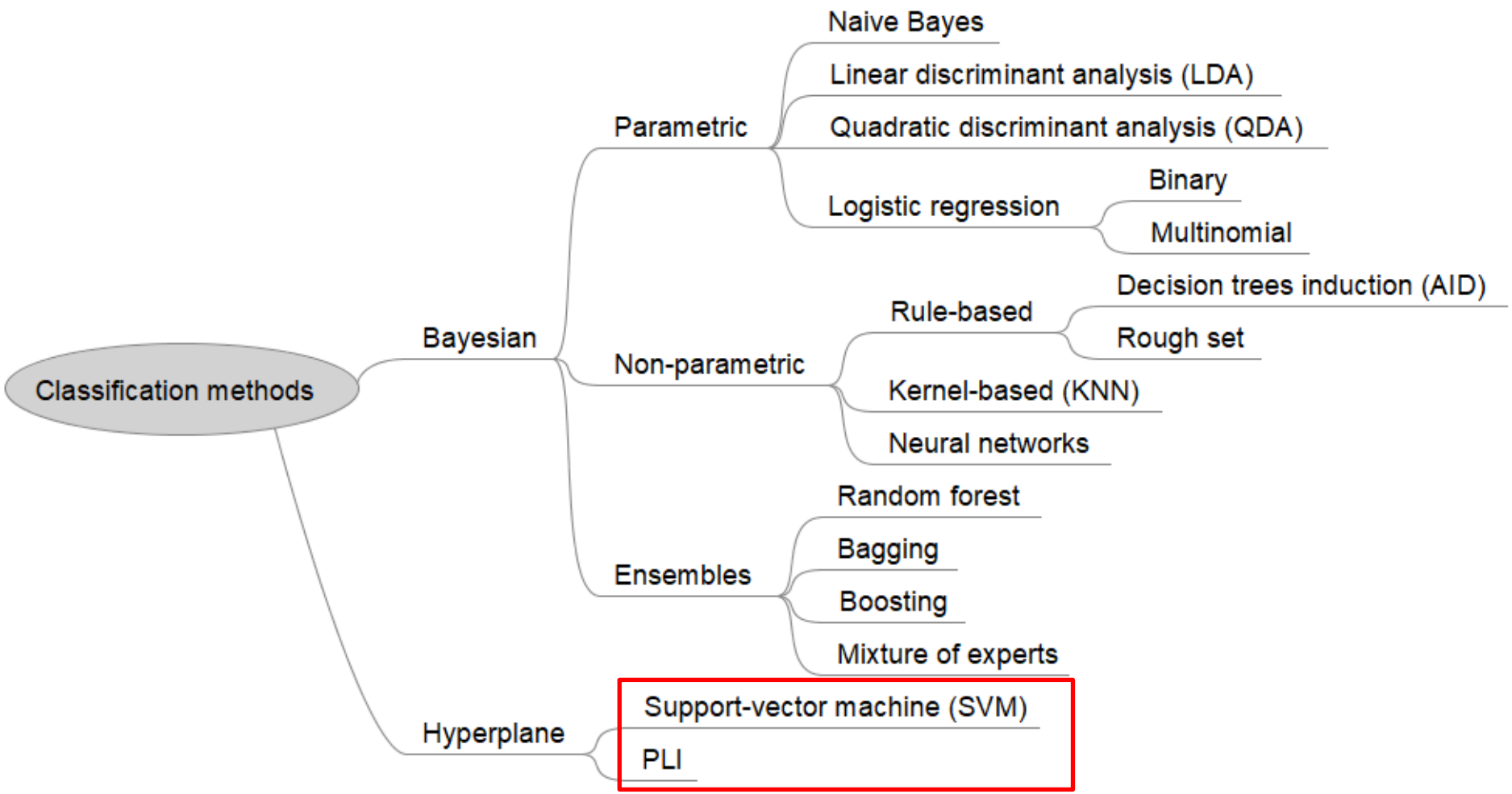


Rodrigo A. Scarpel

rodrigo@ita.br

www.ief.ita.br/~rodrigo

Métodos de classificação:

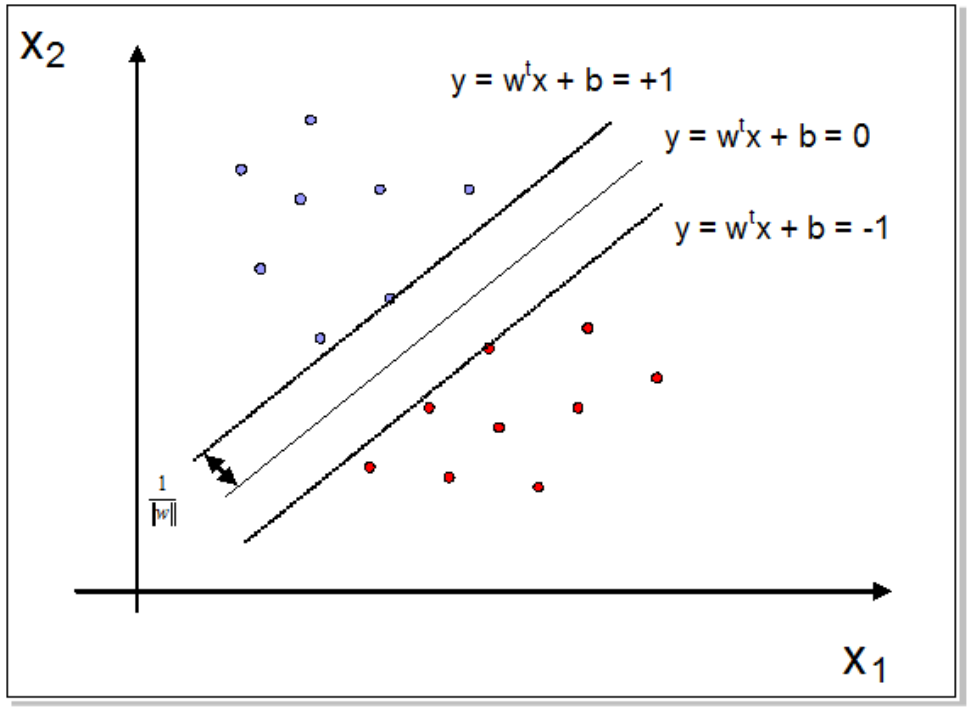


Caso 5: Classificador por Support Vector Machine

Support Vector Machine (SVM) – CLASSIFICAÇÃO BINÁRIA ($y_i \in$ -1,1)

PROBLEMA: achar uma função , linear ou não, para um hiperplano de separação dos pontos em dois conjuntos no R^m, em que m é o número de dimensões existentes.

Caso 1: populações separáveis por um hiperplano linear



- Dados: N observações no R^M (M:dimensões)

  $x_i = x_{i1},\ldots, x_{iM} \quad y_i \in -1,+1 \quad i = 1,\ldots,N$

- Separable Linear Kernel Problem:

  Objetivo: $\text{Maximizar } D(H_{+1}, H_{-1}) = \frac{2}{\|w\|} = \frac{2}{w^t w}$

  $= \text{Minimizar } \frac{1}{D} = \frac{w^t w}{2}$

  Restrições:
  $w^t x + b \geq +1$ para y = +1
  $w^t x + b \leq -1$ para y = -1

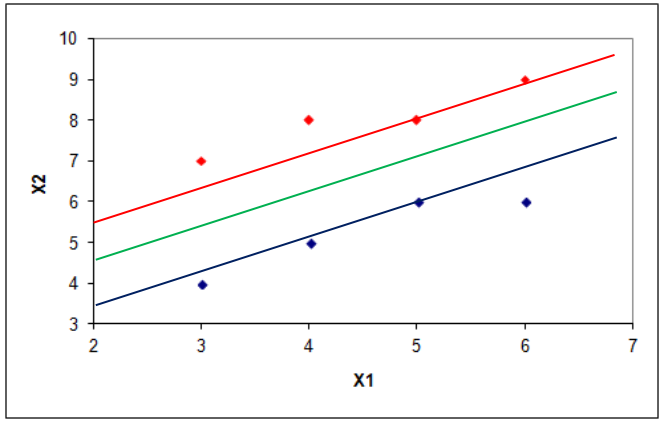
  $\rightarrow \; y(w^t x + b) \geq +1$

Caso 5: Classificador por Support Vector Machine

Support Vector Machine (SVM) – CLASSIFICAÇÃO BINÁRIA ($y_i \in$ -1,1)

Exemplo:

X1	X2	Y
4	5	-1
6	6	-1
3	4	-1
5	6	-1
6	9	+1
4	8	+1
5	8	+1
3	7	+1



Solução:
$w_1 = -0,5$
$w_2 = 1,0$
$b = -4,5$

$$Min\ \frac{1}{2}w^t w = \frac{1}{2}\begin{bmatrix} w_1 & w_2 \end{bmatrix}\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \frac{1}{2}(w_1^{\,2} + w_2^{\,2})$$

s.a. $y(w^t x + b) \geq +1 \longrightarrow y(w_1.x_1 + w_2.x_2 + b) \geq +1$

$-1(w_1.4 + w_2.5 + b) \geq +1$ $\quad +1(w_1.6 + w_2.9 + b) \geq +1$

$-1(w_1.6 + w_2.6 + b) \geq +1$ $\quad +1(w_1.4 + w_2.8 + b) \geq +1$

$-1(w_1.3 + w_2.4 + b) \geq +1$ $\quad +1(w_1.5 + w_2.8 + b) \geq +1$

$-1(w_1.5 + w_2.6 + b) \geq +1$ $\quad +1(w_1.3 + w_2.7 + b) \geq +1$

Hiperplano de Separação (H_0):

$w^t x + b = 0 \rightarrow -0,5.\ x_1 + 1,0.\ x_2 - 4,5 = 0$

$\therefore\ x_2 = 4,5 + 0,5.\ x_1$

Hiperplano Superior (H_{+1}):

$w^t x + b = +1 \rightarrow \therefore\ x_2 = 5,5 + 0,5.\ x_1$

Hiperplano Inferior (H_{-1}):

$w^t x + b = -1 \rightarrow \therefore\ x_2 = 3,5 + 0,5.\ x_1$

Caso 5: Classificador por Support Vector Machine

Support Vector Machine (SVM) – CLASSIFICAÇÃO BINÁRIA ($y_i \in$ -1,1)

Caso 2: populações não separáveis por um hiperplano linear

Modificações:

Introduz-se N variáveis de folga $\xi_i \geq 0$, i=1,…,N

Modifica-se as restrições para:

$w^t x_i + b \geq +1 - \xi_i$ para $y_i = +1$

$w^t x_i + b \leq -1 + \xi_i$ para $y_i = -1$

Combinando as duas restrições:

$y_i(w^t x_i + b) \geq +1 - \xi_i$

Cria-se uma penalidade na função objetivo, obtendo-se um modelo com N + M + 1 incógnitas (ξ_1,…, ξ_N,w_1,…,w_M,b):

$$\text{Min } \frac{1}{2} w^t w + C\left(\sum_{i=1}^{N} \xi_i\right)$$

onde C é uma constante de penalização (C>0)

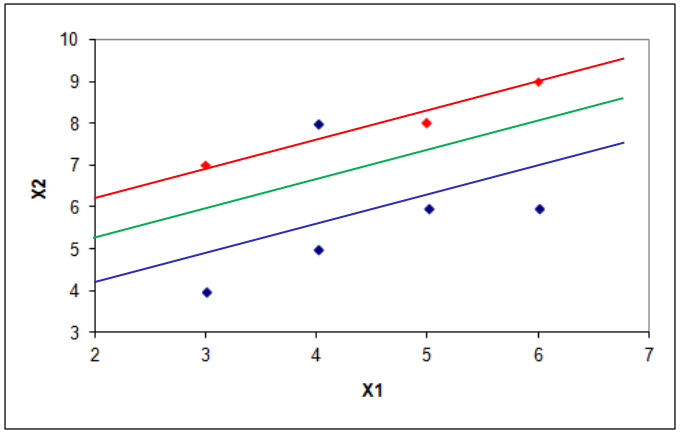
Caso 5: Classificador por Support Vector Machine

Support Vector Machine (SVM) – CLASSIFICAÇÃO BINÁRIA ($y_i \in$ -1,1)

Exemplo:

(C=1)

X1	X2	Y
4	5	-1
6	6	-1
3	4	-1
5	6	-1
6	9	+1
4	8	-1
5	8	+1
3	7	+1



Solução:

w_1 = - 0,5714 ξ_1 = 0

w_2 = 0,8571 ξ_2 = 0

b = - 3,2857 ξ_3 = 0

ξ_4 = 0

ξ_5 = 0

ξ_6 = 2,286

ξ_7 = 0,286

ξ_8 = 0

$$\text{Min } \frac{1}{2}(w_1^{\,2} + w_2^{\,2}) + (\xi_1 + \xi_2 + \xi_3 + \xi_4 + \xi_5 + \xi_6 + \xi_7 + \xi_8)$$

s.a. $y(w_1.x_1+w_2.x_2 + b) \geq +1 - \xi_i$ $\quad \xi_1, \xi_2, \xi_3, \xi_4, \xi_5, \xi_6, \xi_7, \xi_8 \geq 0$

$-1.(w_1.4+w_2.5 + b) \geq +1 - \xi_1$ $\quad +1.(w_1.6+w_2.9 + b) \geq +1 - \xi_5$

$-1.(w_1.6+w_2.6 + b) \geq +1 - \xi_2$ $\quad -1.(w_1.4+w_2.8 + b) \geq +1 - \xi_6$

$-1.(w_1.3+w_2.4 + b) \geq +1 - \xi_3$ $\quad +1.(w_1.5+w_2.8 + b) \geq +1 - \xi_7$

$-1.(w_1.5+w_2.6 + b) \geq +1 - \xi_4$ $\quad +1.(w_1.3+w_2.7 + b) \geq +1 - \xi_8$

Hiperplano de Separação (H_0):

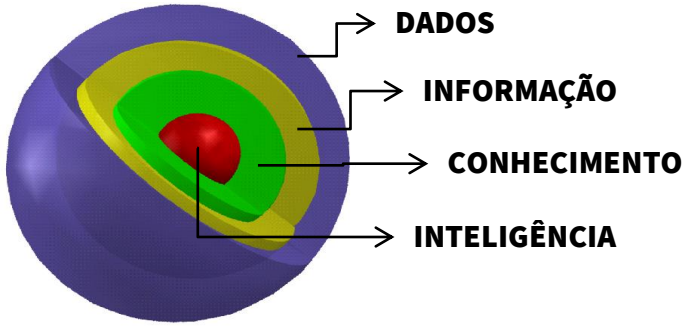
$-0,57.\ x_1 + 0,86.\ x_2 - 3,3 = 0$

$\therefore\ x_2 = 3,83 + 0,67.\ x_1$

(H_{+1}): $x_2 = 4,83 + 0,67.\ x_1$

(H_{-1}): $x_2 = 2,83 + 0,67.\ x_1$

Classificadores CART e mistura de classificadores

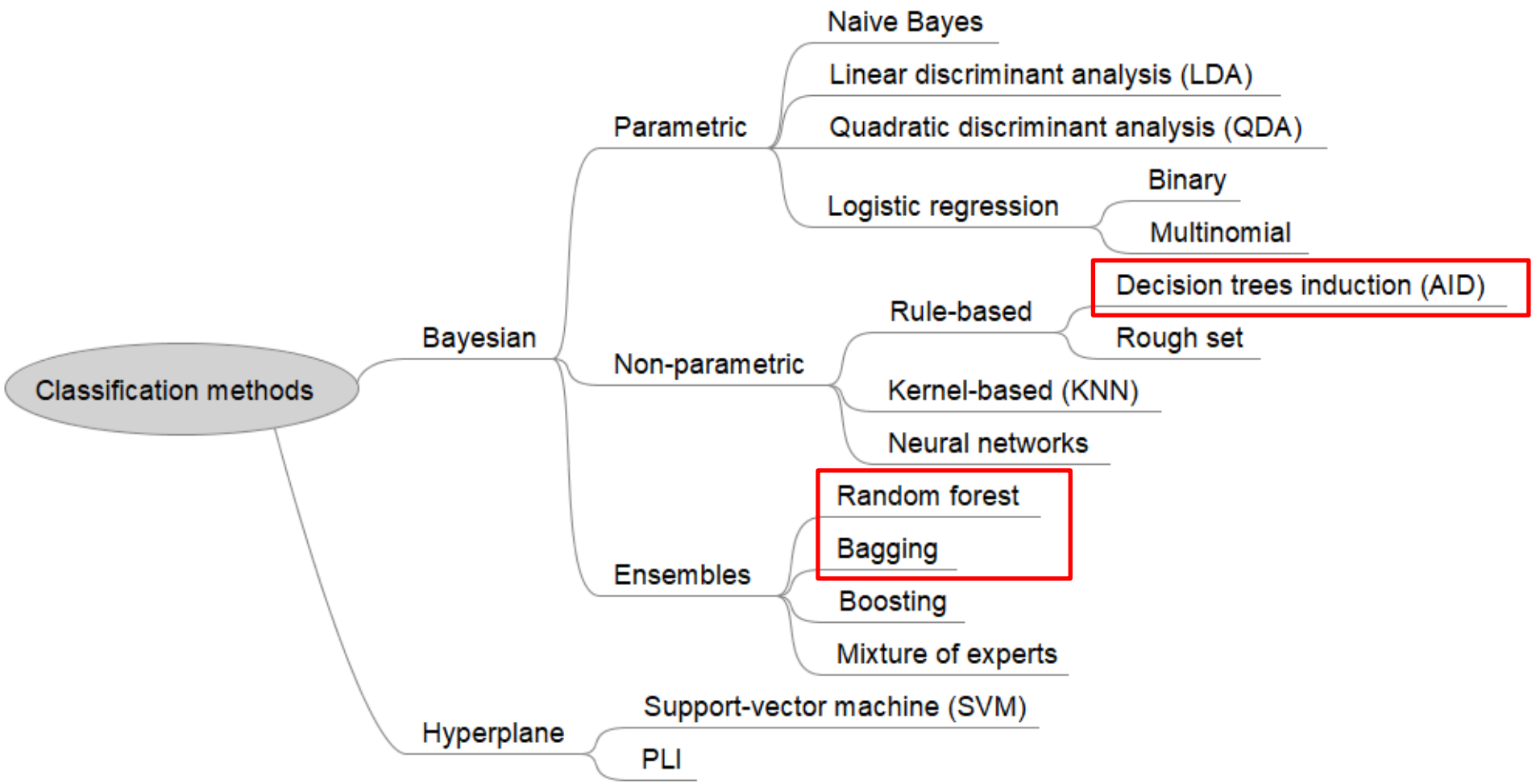


Rodrigo A. Scarpel

rodrigo@ita.br

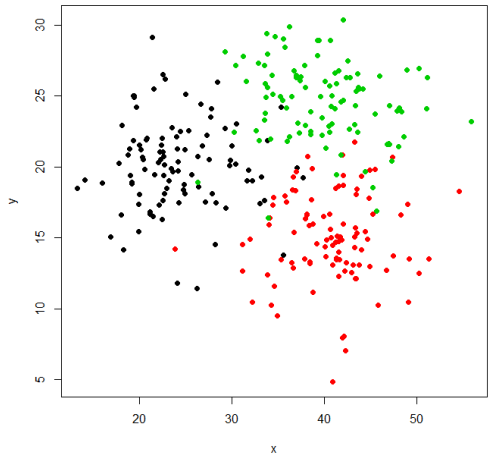
www.ief.ita.br/~rodrigo

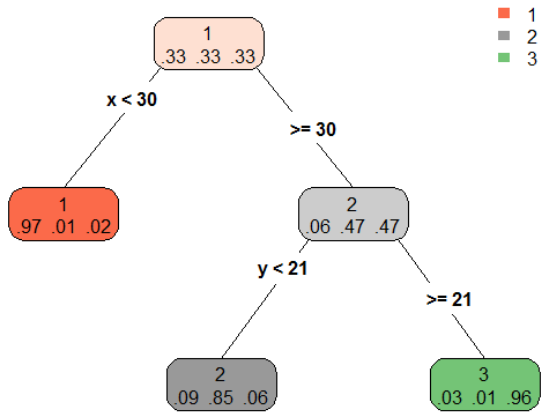
Métodos de classificação:

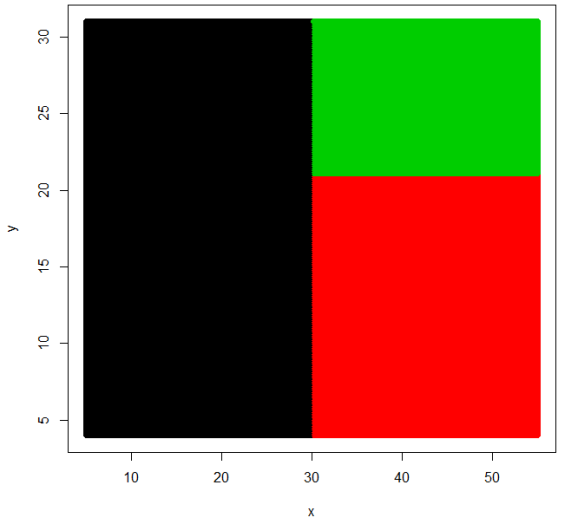


Classificador CART / AID

- Sistema de classificação que particiona o espaço de atributos de forma a criar regras para definir as classes
- Resultado: um conjunto de regras e uma árvore (diagrama)







- Elementos da árvore de decisão
  - Nó raiz (primeira questão)
  - Ramos (possíveis respostas)
  - Outros nós (outras questões)
  - Nó terminal (decisão final)

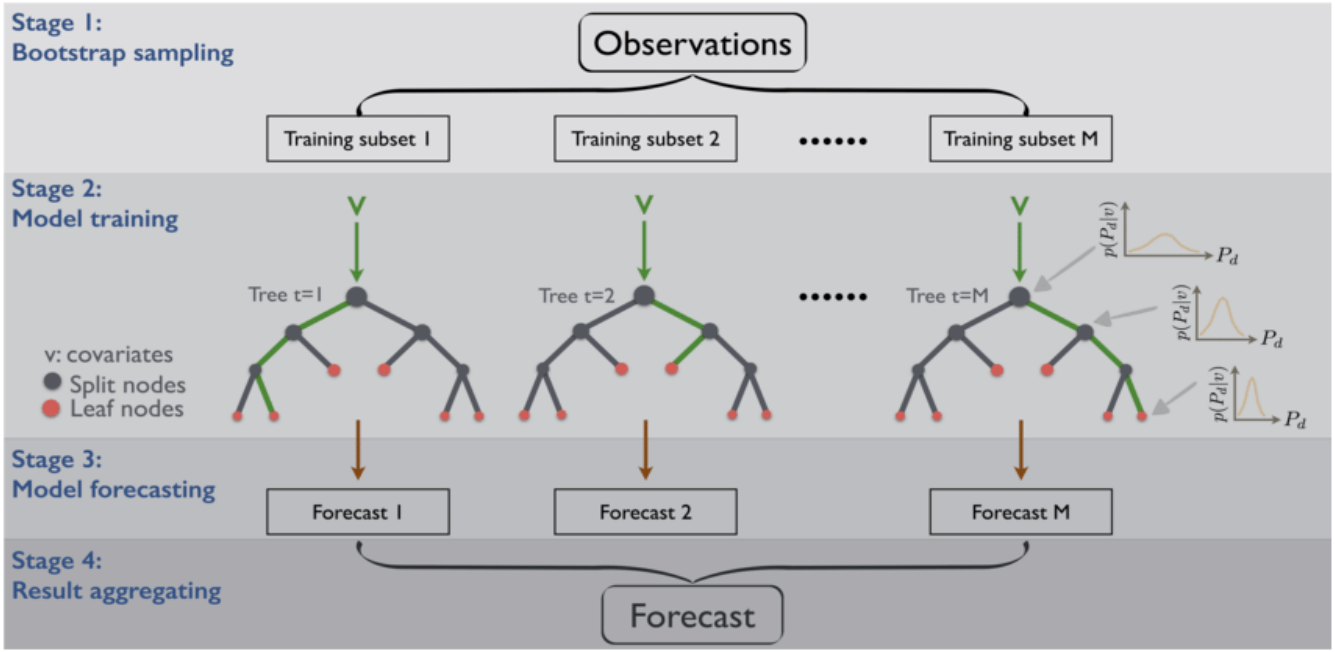
- Critérios de partição:
  - Erro de classificação
  - Índice de Gini: $1 - \sum_{j=1}^{C} p_j^{\,2}$

Mistura de classificadores (Ensemble models)

- **Bagging (Bootstrapp AGGregatING):**

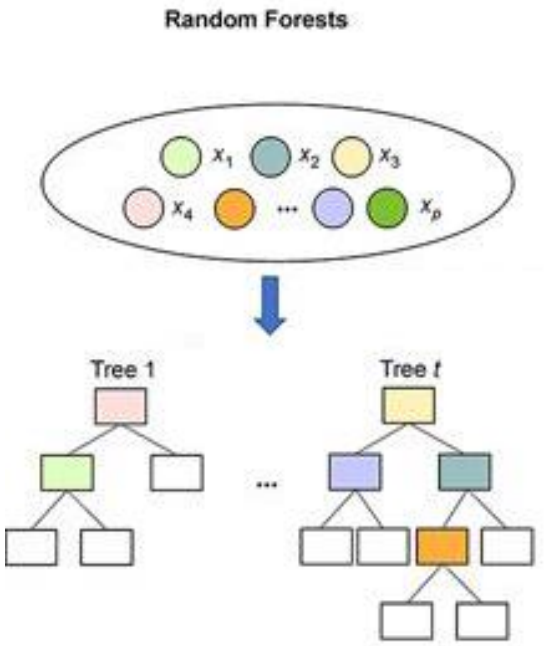
  No estágio 1, cria-se amostras aleatórias de conjuntos de treinamento (com reposição), no estágio 2 cria-se um classificador (CART) para cada conjunto de treinamento e nos estágios 3 e 4 combina-se a previsão dos modelos obtidos (utilizando a média ou a moda dos resultados).

OBS: O Bagging auxilia na redução da variância do erro de previsão / classificação



Mistura de classificadores (Ensemble models)

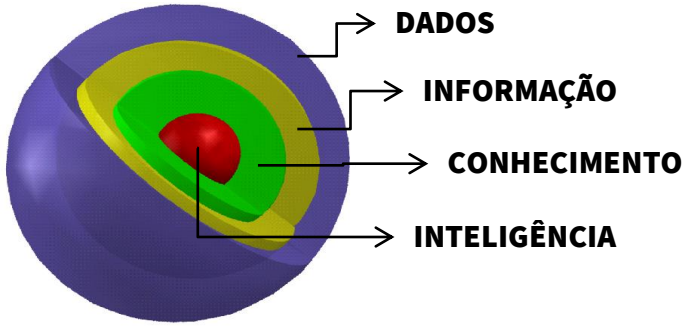
- **Random Forest:**



A Random Forest faz uso do mesmo método do Bagging, ou seja, no estágio 1 cria-se amostras aleatórias de conjuntos de treinamento (com reposição), no estágio 2 cria-se um classificador (CART) para cada conjunto de treinamento e nos estágios 3 e 4 combina-se a previsão dos modelos obtidos (utilizando a média ou a moda dos resultados).

A diferença é que em cada classificador (CART) um subconjunto das variáveis independentes candidatas para compor o classificador são aleatóriamente escolhidas (há um número máximo de variáveis default).

Avaliação de classificadores e práticas em classificação



Rodrigo A. Scarpel

rodrigo@ita.br

www.ief.ita.br/~rodrigo

Avaliação de classificadores

Há muitas métricas de avaliação de modelos de classificação. As mais comumente encontradas são:

- Taxa de acerto (acurácia): percentual de observações corretamente classificadas

  (= TP + TN / TP + FP + FN + TN)

- Taxa de erro = 1 – Taxa de acerto = FN + FP / TP + FP + FN + TN

- Sensitivity (= Recall ou TPR): percentual de reais Ps corretamente classificados

  = TP / TP + FN

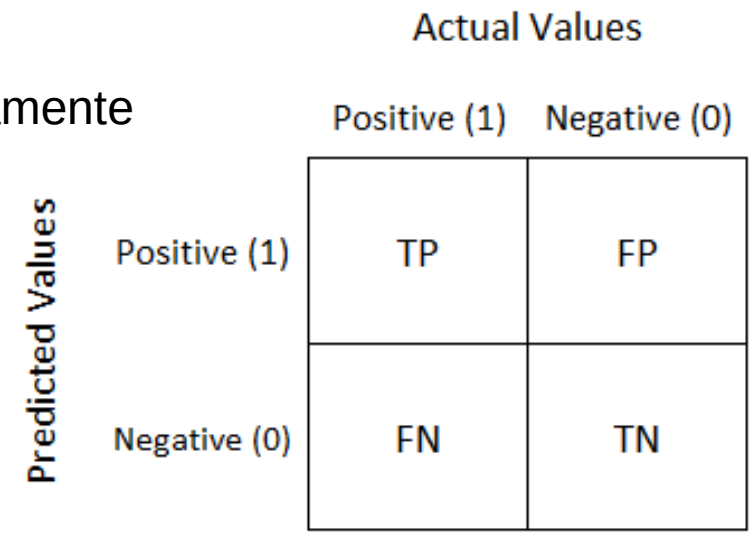
- Specificity (ou TNR): percentual de reais Ns corretamente

  classificados = TN / TN + FP)

- Precision (ou PPV): percentual de classificados

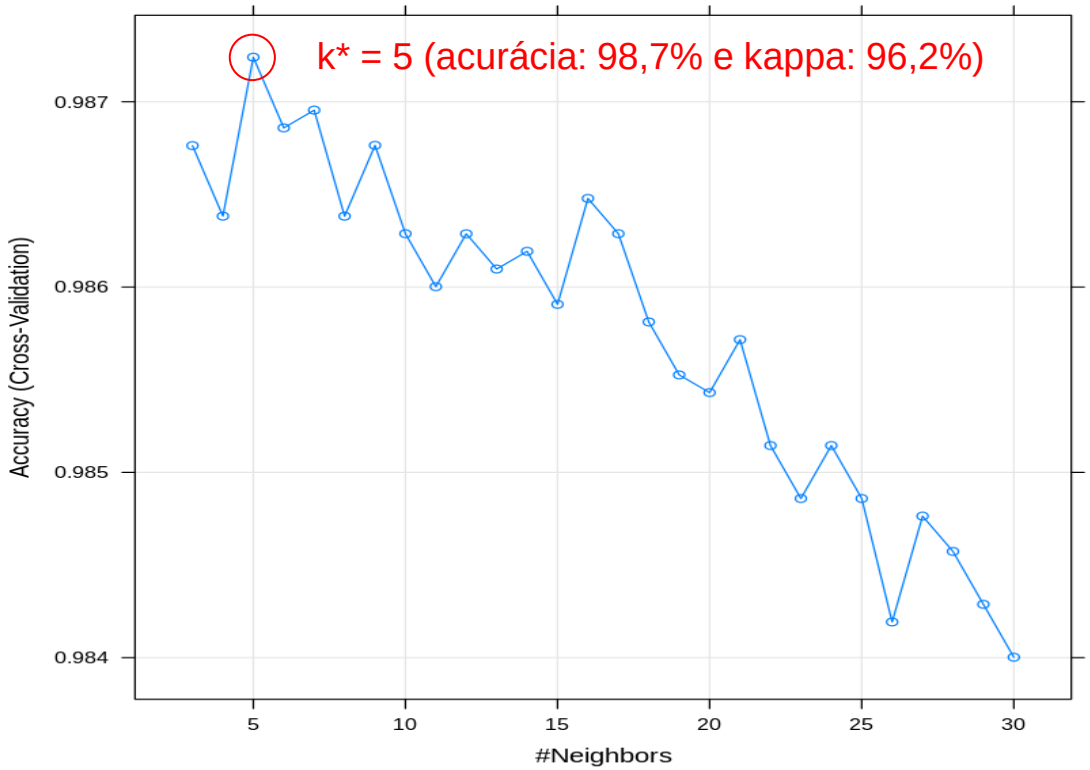
  como P que realmente são P

  = TP / TP + FP



Avaliação de classificadores

Acurácia e Kappa:



$$\text{Acurácia} = \frac{TP + TN}{TP + TN + FP + FN}$$

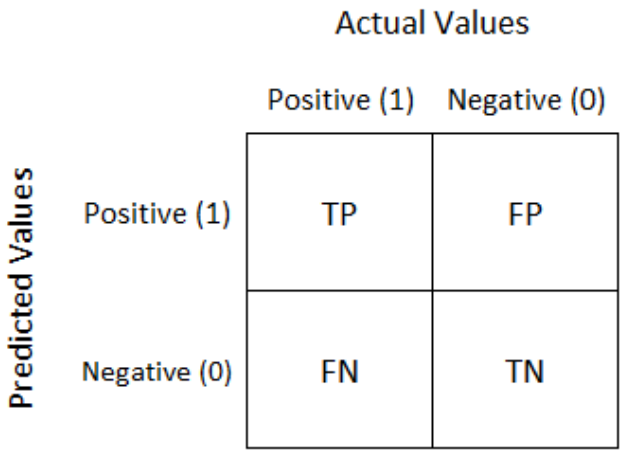
$$\text{Kappa} = 1 - \frac{1 - \text{Acurácia}}{1 - \text{Priori}}$$

		Predicted class	
		P	N
Actual Class	P	True Positives (TP)	False Negatives (FN)
	N	False Positives (FP)	True Negatives (TN)

OBS: Priori = $(8.327/10.501)^2 + (2.174/10.501)^2 = 67{,}17\%$

Avaliação de classificadores

- **No exemplo:**



Accuracy = (TP + TN) / TOTAL

Sensitivity = TP / (TP + FN)

Specificity = TN / (TN + FP)

PPV (ou Precision) = TP / (TP+FP)

NPV = TN / (TN + FN)

Prevalence = (TP + FN) / TOTAL

Detection Rate = TP / TOTAL

Det. Prevalence = (TP + FP) / TOTAL

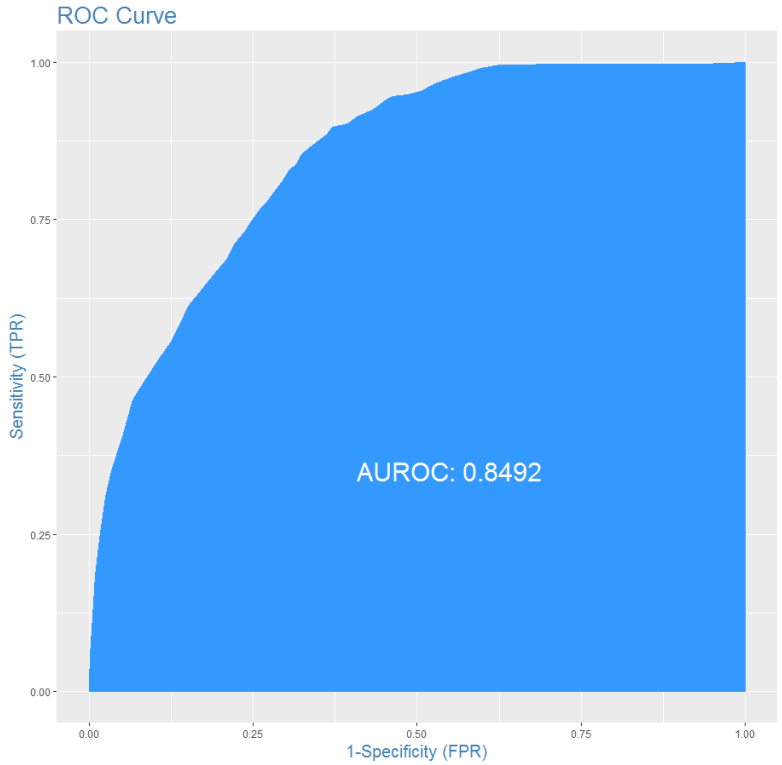
Balanced Acc. = (sensitivity + specificity)/2

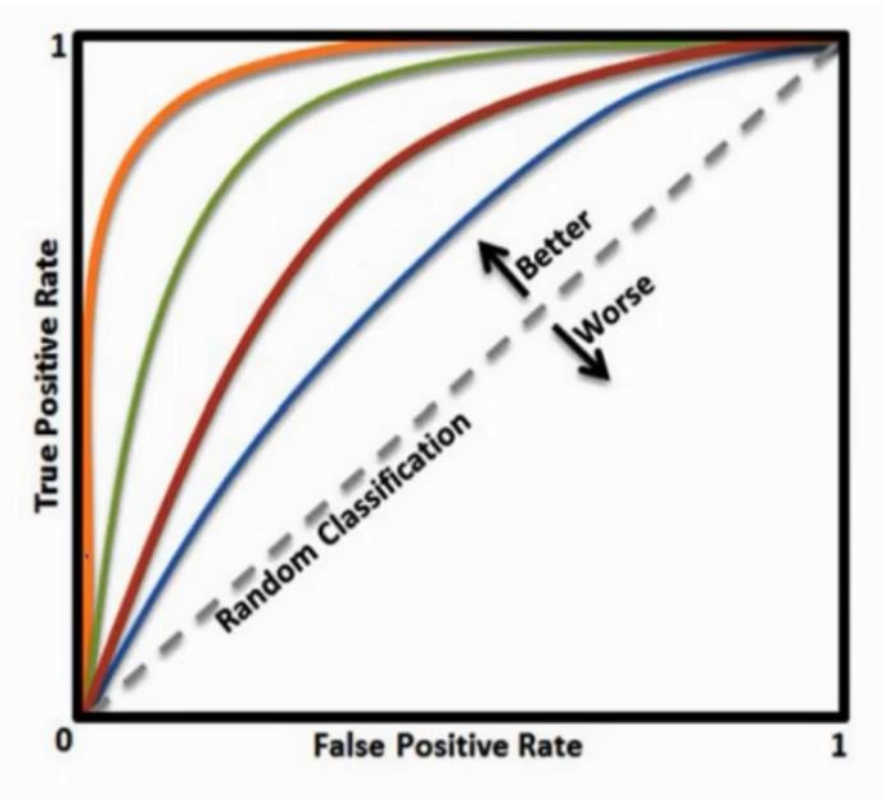
```
Confusion Matrix and Statistics

          Reference
Prediction    N    S
         N 3333  235
         S  193  738

               Accuracy : 0.9049
                 95% CI : (0.8959, 0.9133)
    No Information Rate : 0.7837
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.7149

 Mcnemar's Test P-Value : 0.0475

            Sensitivity : 0.9453
            Specificity : 0.7585
         Pos Pred Value : 0.9341
         Neg Pred Value : 0.7927
             Prevalence : 0.7837
         Detection Rate : 0.7408
   Detection Prevalence : 0.7931
      Balanced Accuracy : 0.8519

       'Positive' Class : N
```

Avaliação de classificadores

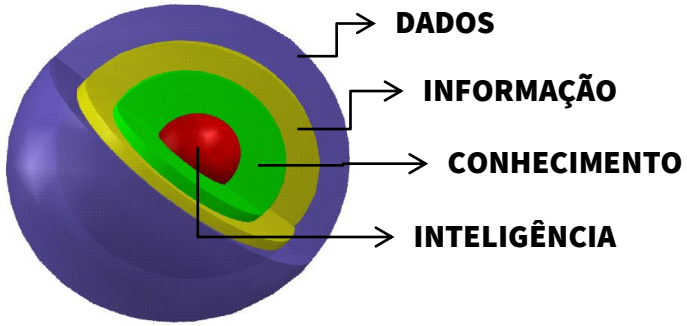
Há muitas métricas de avaliação de modelos de classificação. As mais comumente encontradas são:

- Curva ROC (1- Specificity x Sensitivity ou FPR x TPR)
- AUROC: área abaixo da curva ROC





Construção de modelos de Regressão

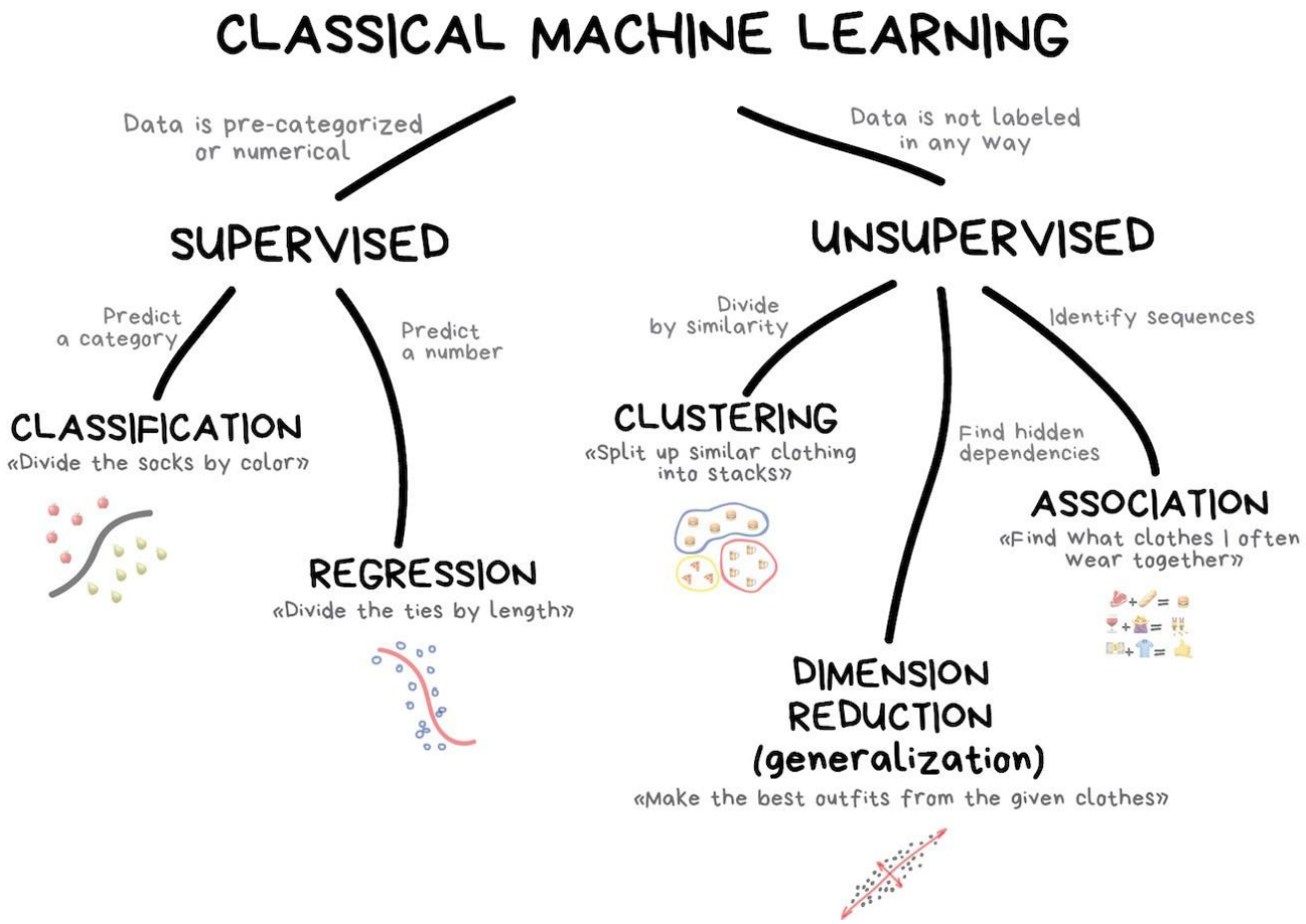


Rodrigo A. Scarpel

rodrigo@ita.br

www.ief.ita.br/~rodrigo

Introdução:



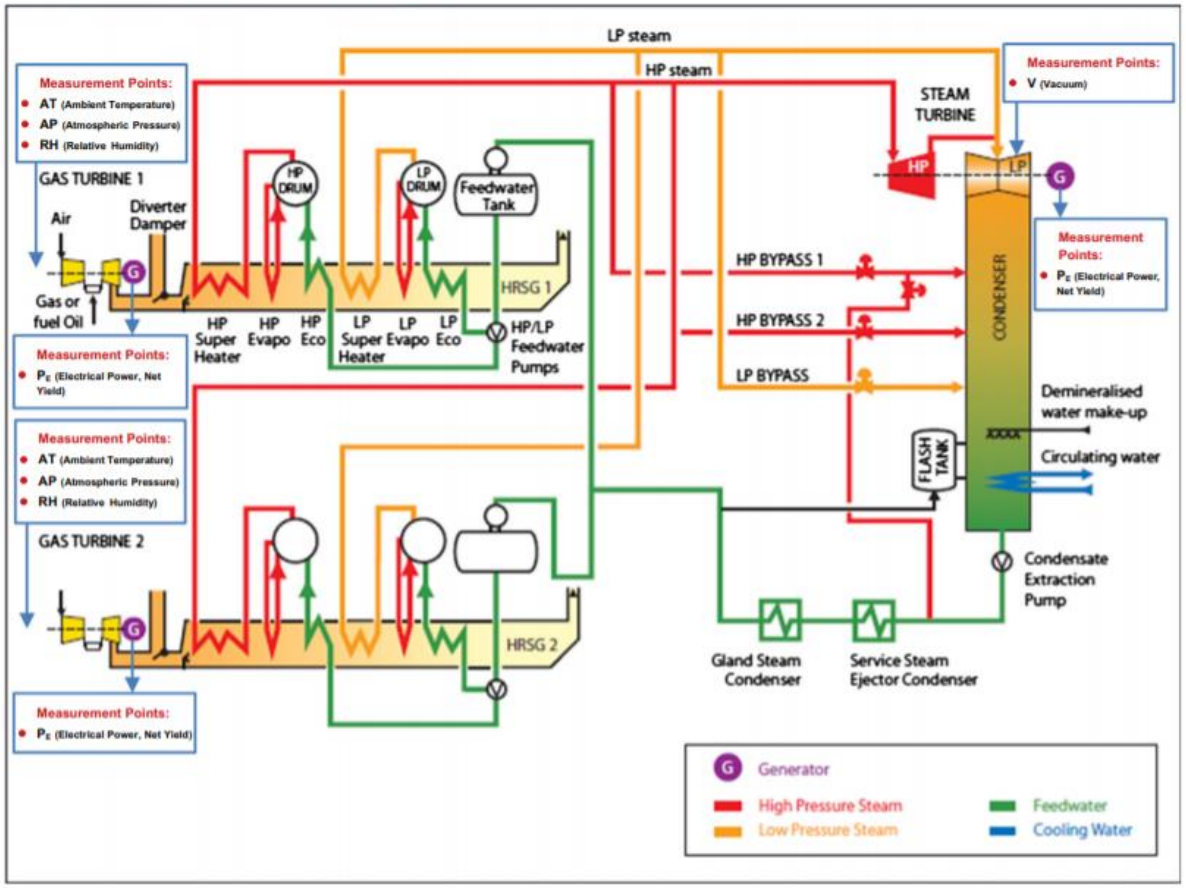
Classificação – problemas em que a variável resposta é categórica
(ex: y = fraude / não fraude
y = número 2 / número 7
y = falha / não falha
y = solvente / insolvente)

Regressão – problemas em que a variável resposta é um número real
(ex: y = vendas mensais
y = energia gerada
y = tempo de ciclo
y = retorno diário do BVSP)

Fonte: https://vas3k.com/blog/machine_learning/

Métodos de Regressão:

- **Previsão de produção de energia (termoelétrica):**



Dados: 9.568 observações

- Variável dependente (Y):

PE: Eletrical Power Yield

- Variáveis independentes (Xs):

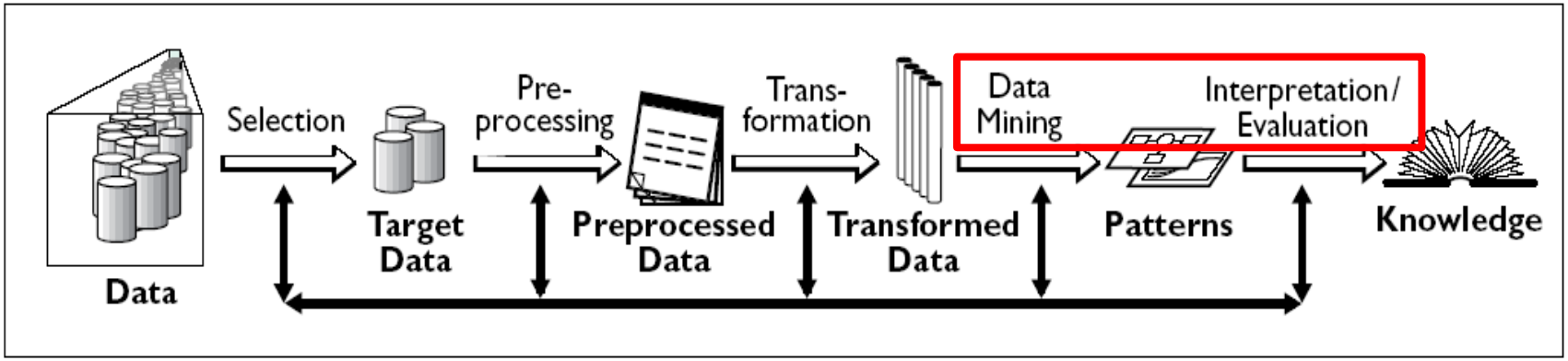
AT : Ambient Temperature

V: Exhaust Vacuum Pressure

AP: Atmosferic Pressure

RH: Relative Humidity

Construção do modelo de regressão:



DADOS DISPONÍVEIS PARA ANÁLISE

DADOS - TREINAMENTO

DADOS - TESTE → Acurácia: Teste (R^2)

5–fold Cross-Validation (CV)

T : treino
V: validação

1:	T	T	T	T	V	→ Acurácia: V_1 (R^2)
2:	T	T	T	V	T	→ Acurácia: V_2 (R^2)
3:	T	T	V	T	T	→ Acurácia: V_3 (R^2)
4:	T	V	T	T	T	→ Acurácia: V_4 (R^2)
5:	V	T	T	T	T	→ Acurácia: V_5 (R^2)

Acurácia: média ($\bar{R}^2$)

Análise de regressão:

- Variáveis dependente e independente:

Variável Dependente (Y)	Variável Independente (X)
Representa a variável do problema que se deseja explicar (precisa ser métrica – escala de intervalo ou de razão)	Explica a variável dependente (uma ou mais variáveis e são não métricas – categóricas com c categorias)
Exemplos: Preferência por uma marca de xampu Volume de consumo de um produto Taxa de retorno mensal de um ativo	Exemplos: Categoria de usuário (assíduo, ...) Segmento (solteiros, ...) Taxa de retorno mensal do mercado

Análise de regressão:

- Análise de Regressão lida com a investigação da relação entre duas ou mais variáveis. É utilizada para:

1. Verificar se existe relação (se as variáveis independentes explicam a variação no comportamento da variável dependente);

2. Intensidade do relacionamento (quanto da variação da variável dependente pode ser explicada pelas variáveis independentes);

3. Determinar a estrutura (ou a forma da relação), ou seja, a equação matemática que relaciona as variáveis independentes com a variável dependente;

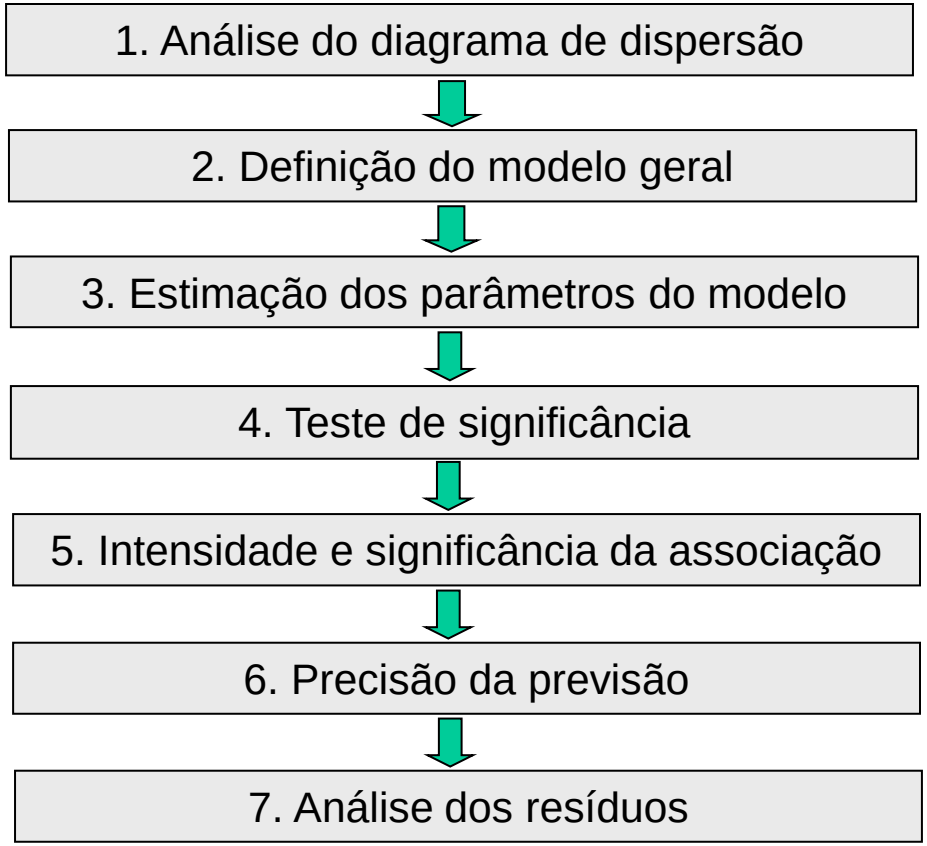
4. Fazer previsões (aplicar a equação matemática obtida para realizar análises do tipo se – então);

5. Fazer prescrições (propor ações de controle das variáveis independentes conforme um valor desejado para a variável dependente)

Análise de regressão:

- Regressão bidimensional: caso em que há apenas 1 variável independente

  Etapas na condução da análise:



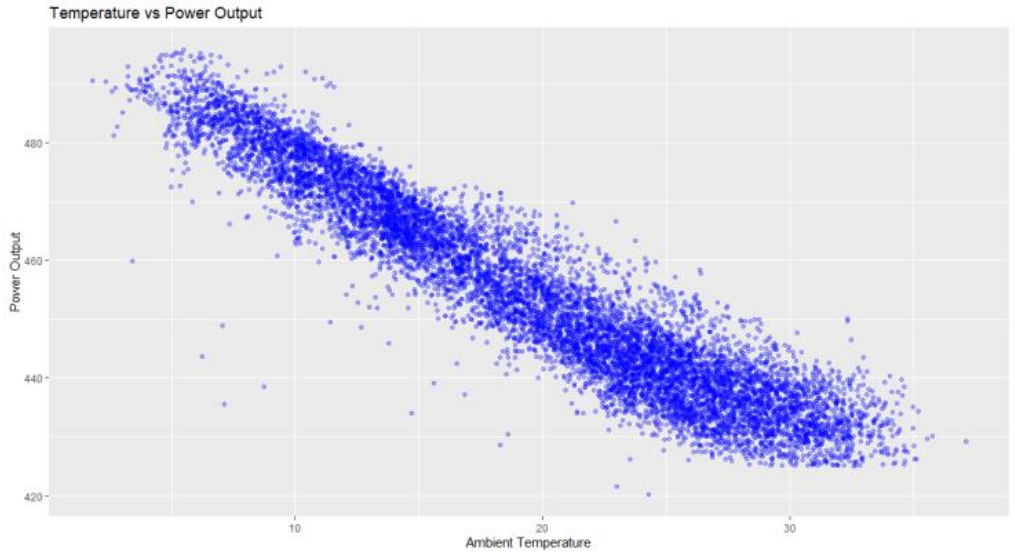
Análise de regressão: caso bidimensional

- Etapas na condução da análise:

1. Análise do diagrama de dispersão
2. Definição do modelo geral
3. Estimação dos parâmetros do modelo
4. Teste de significância
5. Intensidade e significância da associação
6. Precisão da previsão
7. Análise dos resíduos

1. Análise do diagrama de dispersão:

   Gráfico dos valores das duas variáveis para todas as observações (variável dependente no eixo horizontal, variável independente no eixo vertical)



Análise de regressão: caso bidimensional

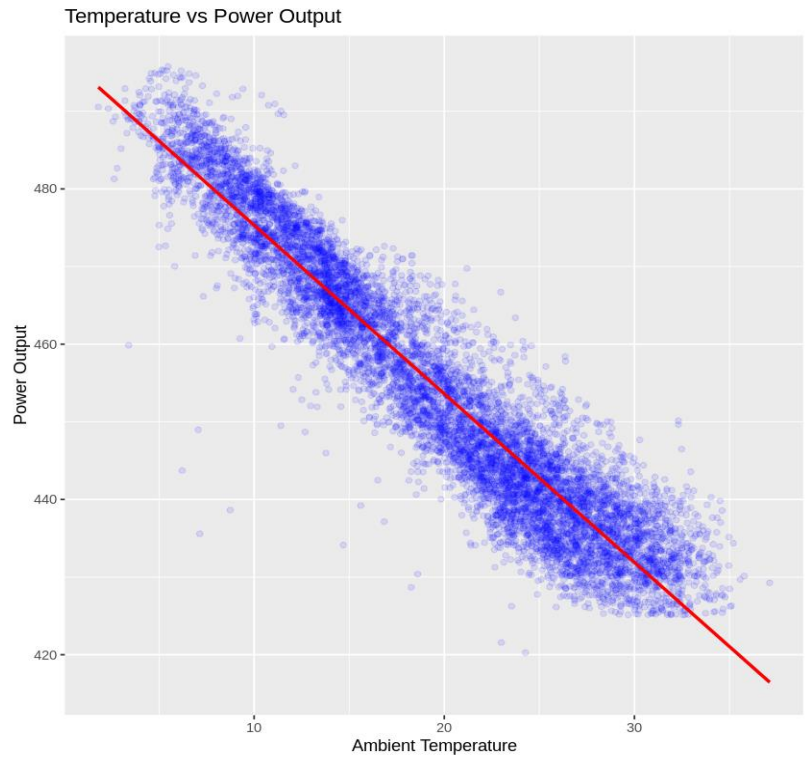
- Etapas na condução da análise:

1. Análise do diagrama de dispersão
2. Definição do modelo geral
3. Estimação dos parâmetros do modelo
4. Teste de significância
5. Intensidade e significância da associação
6. Precisão da previsão
7. Análise dos resíduos

2. Definição do modelo geral:

Linear (reta): $Y = \beta_0 + \beta_1.X + \varepsilon$

intercepto ↲ ↳ coeficiente angular



Análise de regressão: caso bidimensional

- Etapas na condução da análise:

1. Análise do diagrama de dispersão
2. Definição do modelo geral
3. Estimação dos parâmetros do modelo
4. Teste de significância
5. Intensidade e significância da associação
6. Precisão da previsão
7. Análise dos resíduos

3. Estimação dos parâmetros do modelo:

$$\text{MQO : Minimizar } \sum_i \left(Y_i - \hat{Y}_i\right)^2$$

Para o caso linear (reta):

$$\hat{\beta}_0 = \overline{Y} - \hat{\beta}_1 \overline{X}$$

$$\hat{\beta}_1 = \frac{\sum_i \left(X_i - \overline{X}\right)\left(Y_i - \overline{Y}\right)}{\sum_i \left(X_i - \overline{X}\right)^2} = \frac{Cov_{X,Y}}{S_X^2}$$

4. Teste de significância:

H_0: $\beta_1 = 0$ (não há relação linear)

H_a: $\beta_1 \neq 0$ (há relação positiva ou negativa)

Estatística do teste: $t = \dfrac{\hat{\beta}_1 - 0}{\hat{\sigma}_{\hat{\beta}_1}}$

Valor crítico: $t_{\alpha/2,n-2}$

Análise de regressão: caso bidimensional

- Etapas na condução da análise:

1. Análise do diagrama de dispersão
2. Definição do modelo geral
3. Estimação dos parâmetros do modelo
4. Teste de significância
5. Intensidade e significância da associação
6. Precisão da previsão
7. Análise dos resíduos

5. Intensidade e significância da associação:

Coeficiente de Determinação (R^2)

$$R^2 = \frac{\sum_i \left(\hat{Y}_i - \overline{Y}\right)^2}{\sum_i \left(Y_i - \overline{Y}\right)^2} = \frac{\text{SQRegressão}}{\text{SQTotal}} = 1 - \frac{\text{SQResíduos}}{\text{SQTotal}}$$

Em regressão linear simples, $R^2 = r_{X,Y}^2$

Significância:

H_0: $R^2_{pop} = 0$ (não há associação)

H_a: $R^2_{pop} > 0$ (há associação significativa)

- Estatística do teste: $F = \frac{\text{SQRegressão}/1}{\text{SQResíduos}/(n-2)}$
- Valor crítico: $F_{\alpha,1,n-2}$

Análise de regressão: caso bidimensional

- Etapas na condução da análise:

1. Análise do diagrama de dispersão
2. Definição do modelo geral
3. Estimação dos parâmetros do modelo
4. Teste de significância
5. Intensidade e significância da associação
6. Precisão da previsão
7. Análise dos resíduos

6. Precisão da previsão:

$$\hat{\sigma} = \sqrt{\frac{\text{SQResíduos}}{n-2}} = \sqrt{\text{QMResíduos}}$$

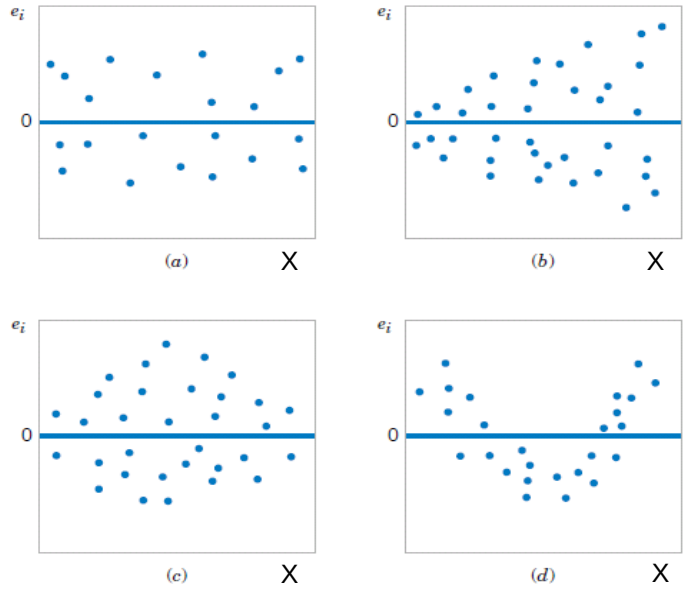
7. Análise dos resíduos:

(a) Padrão ideal

(b) Variância não é constante

(c) Variância não é constante

(d) Resíduos não são independentes

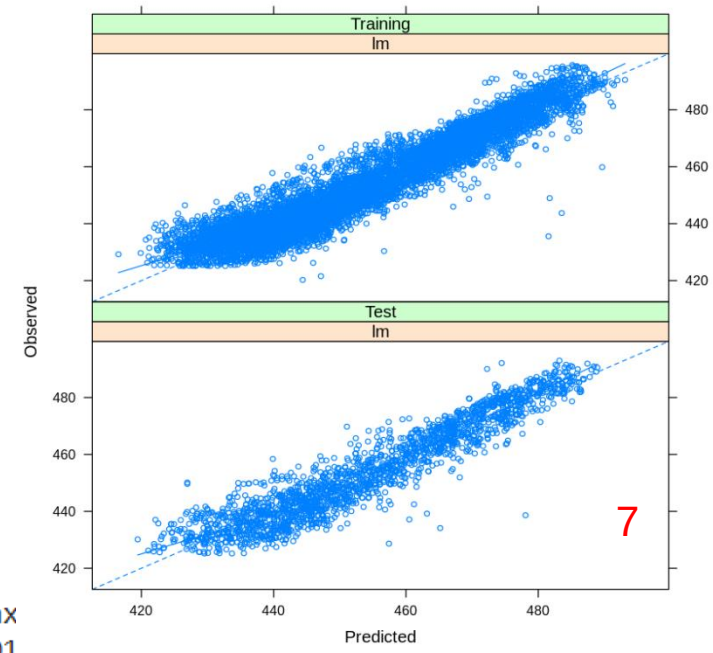


Análise de regressão: caso bidimensional

- Exemplo:

1. Análise do diagrama de dispersão √
2. Definição do modelo geral √
3. Estimação dos parâmetros do modelo
4. Teste de significância
5. Intensidade e significância da associação
6. Precisão da previsão
7. Análise dos resíduos

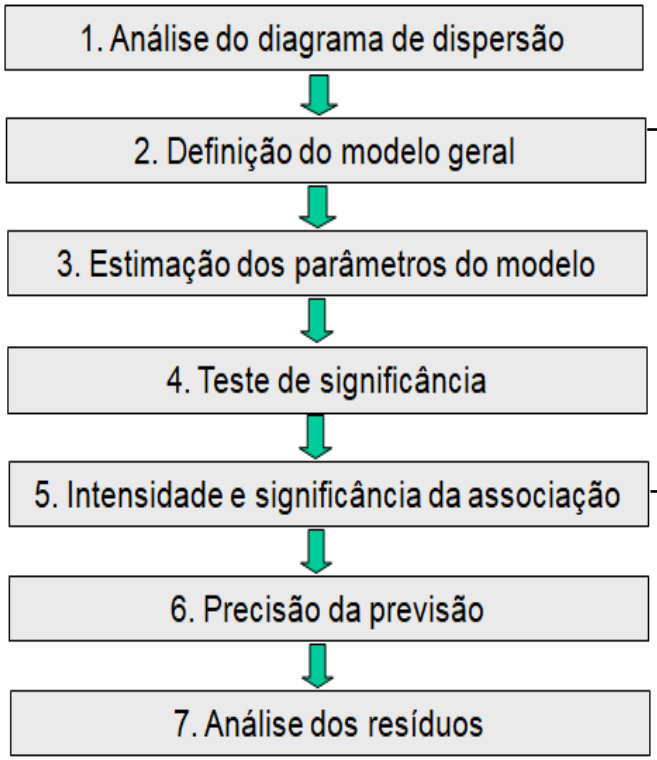
```
Call:
lm(formula = .outcome ~ ., data = dat)

Residuals:
    Min      1Q  Median      3Q     Max
-45.973  -3.648   0.110   3.660  19.901

Coefficients:                                          3                4
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 497.042508   0.172319  2884.4   <2e-16 ***
AT           -2.169454   0.008193  -264.8   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

6 Residual standard error: 5.359 on 7654 degrees of freedom
5 Multiple R-squared:  0.9016,    Adjusted R-squared:  0.9016
F-statistic: 7.012e+04 on 1 and 7654 DF,  p-value: < 2.2e-16
```



Análise de regressão:

- Regressão múltipla: caso em que há mais de 1 variável independente

Linear (reta): $Y = \beta_0 + \beta_1.X_1 + \beta_2.X_2 + \beta_3.X_3 + \ldots + \beta_k.X_k + \varepsilon$



2. Definição do modelo geral:

Seleção das variáveis do modelo: Backward elimination

5. Intensidade e significância da associação:

Coeficiente de Determinação Ajustado ($R^2_{ajustado}$)

$$R^2_{ajustado} = R^2 \frac{k(1-R^2)}{n-k-1}$$

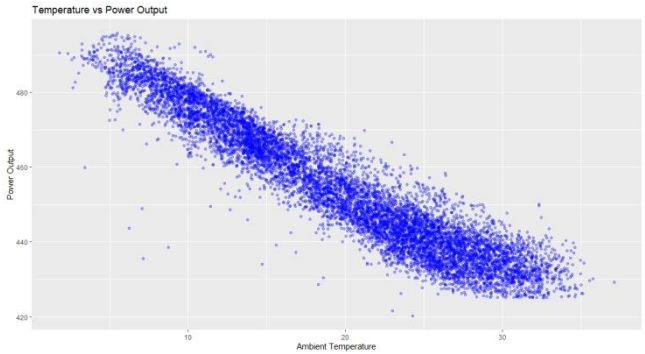
Significância: H_0: $\beta_1 = \beta_2 = \beta_3 = \ldots \beta_k = 0$

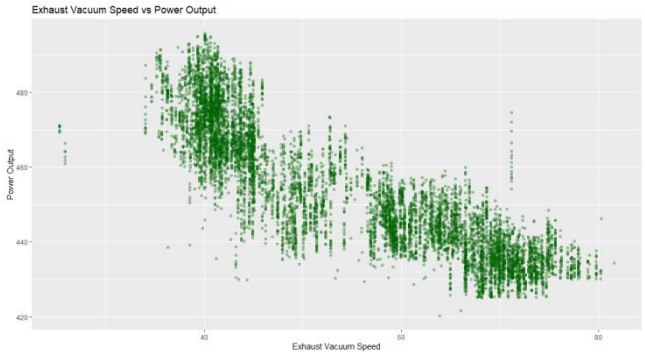
H_a: algum $\beta_i \neq 0$ (i = 1,...,k)

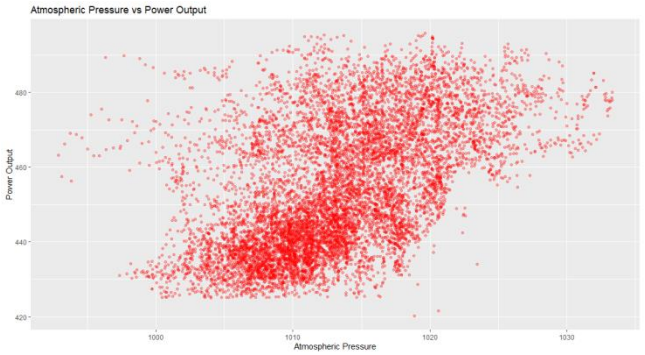
- Estatística do teste: $F = \dfrac{SQReg/k}{\dfrac{SQRes}{(n-k-1)}} = \dfrac{R^2/k}{\dfrac{(1-R^2)}{(n-k-1)}}$
- Valor crítico: $F_{\alpha,k,n-k-1}$

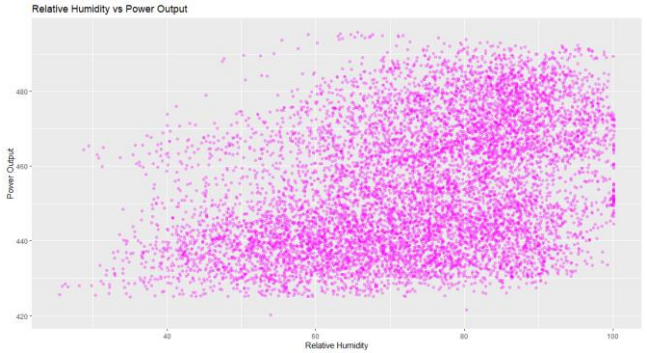
Métodos de Regressão:

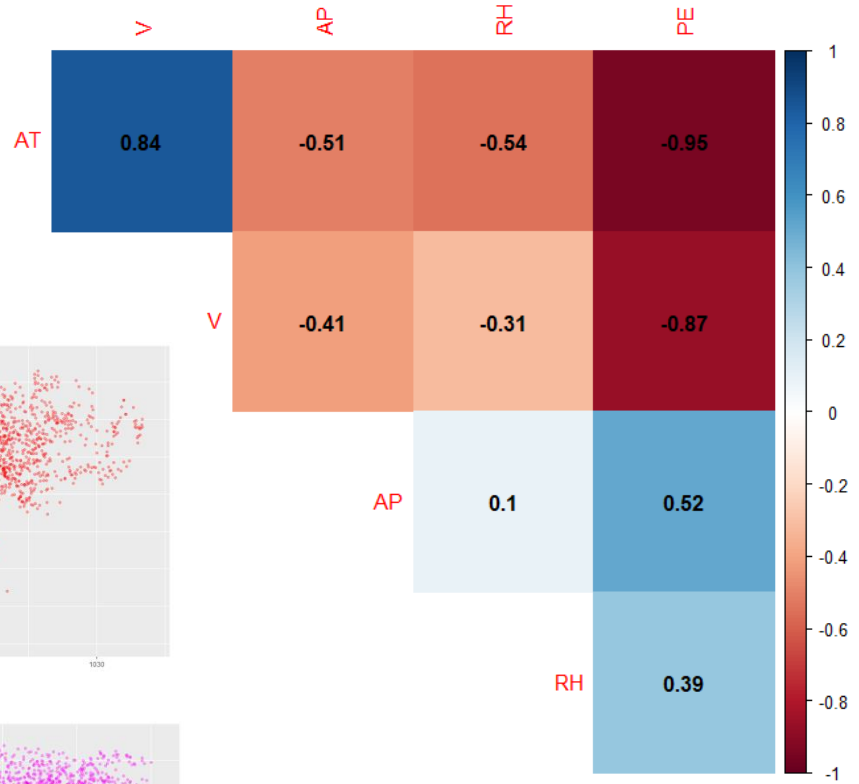
- Análise exploratória dos dados:





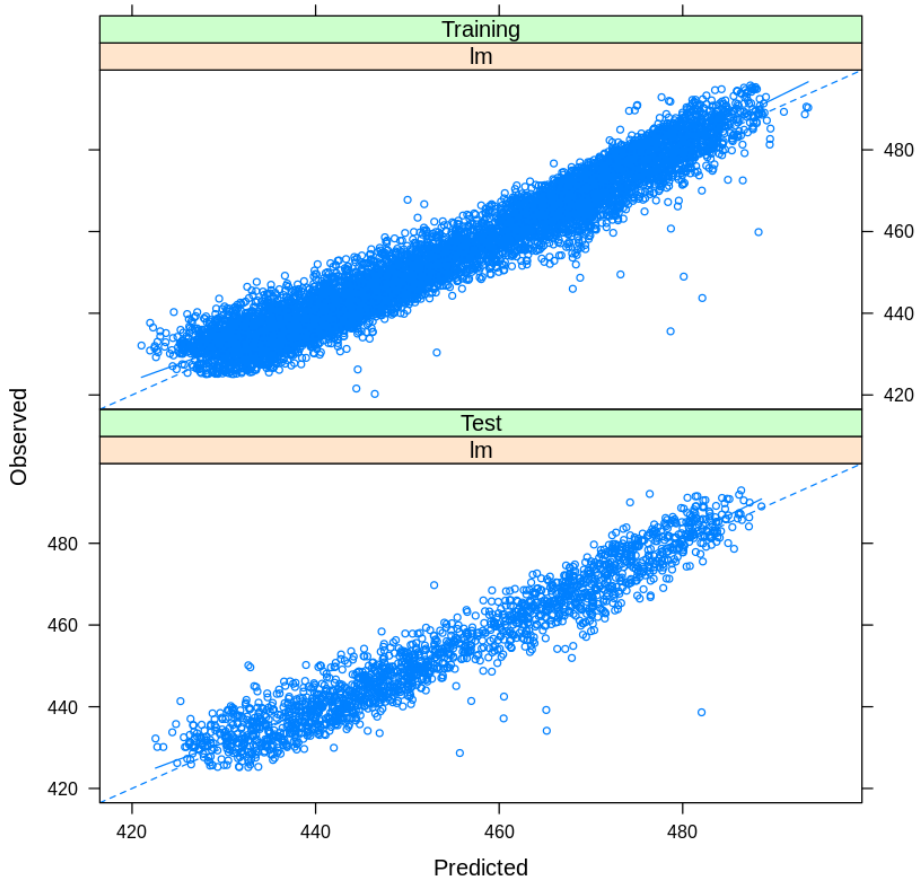






Regressão Linear Múltipla:

```
Linear Regression

7656 samples
   4 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 6891, 6890, 6891, 6892, 6890, 6890, ...
Resampling results:

  RMSE      Rsquared   MAE
  4.504955  0.9306019  3.606429

Tuning parameter 'intercept' was held constant at a value of TRUE

Call:
lm(formula = .outcome ~ ., data = dat)

Residuals:
    Min      1Q  Median      3Q     Max
-43.101  -3.146  -0.132   3.173  17.735

Coefficients:
             Estimate Std. Error  t value Pr(>|t|)
(Intercept) 453.607920  10.778493   42.085  < 2e-16 ***
AT           -1.984670   0.016884 -117.544  < 2e-16 ***
V            -0.227863   0.008049  -28.308  < 2e-16 ***
AP            0.062865   0.010456    6.012 1.91e-09 ***
RH           -0.157308   0.004596  -34.231  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.504 on 7651 degrees of freedom
Multiple R-squared:  0.9305,    Adjusted R-squared:  0.9305
F-statistic: 2.561e+04 on 4 and 7651 DF,  p-value: < 2.2e-16
```

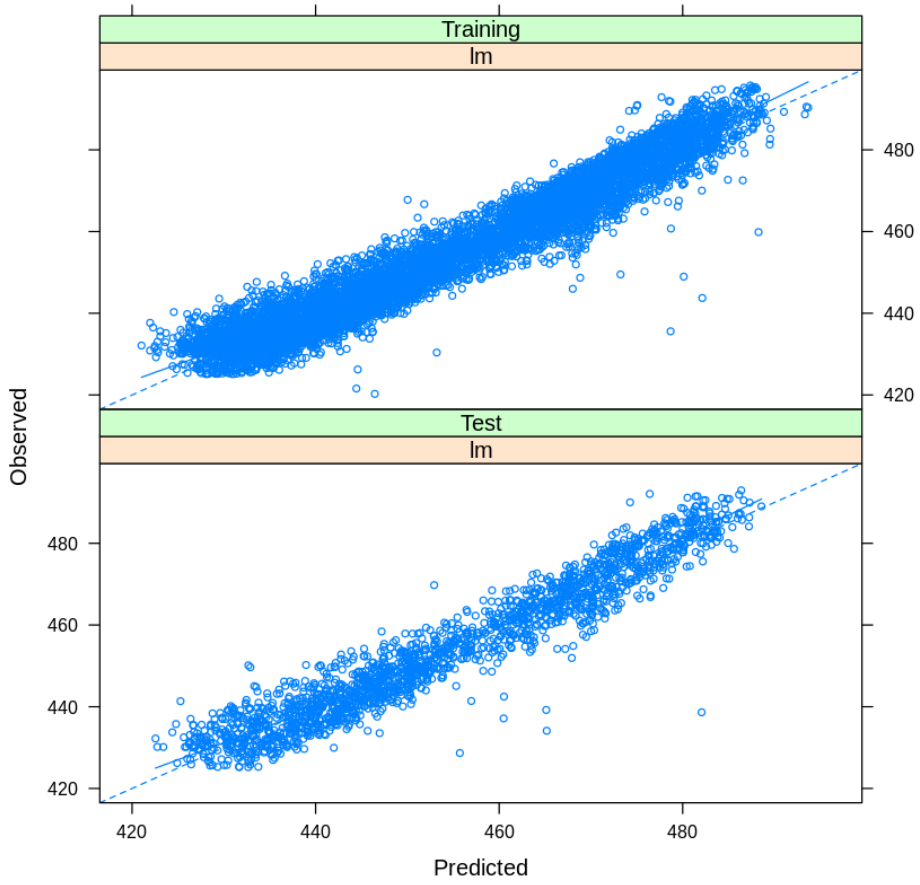


No conjunto de teste:

RMSE	Rsquared	MAE
4.769306	0.9214907	3.71388

Regressão Linear Múltipla:

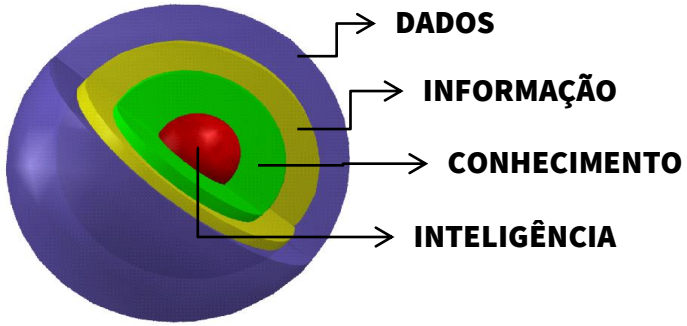
```
Linear Regression

7656 samples
   4 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 6891, 6890, 6891, 6892, 6890, 6890, ...
Resampling results:

  RMSE      Rsquared   MAE
  4.504955  0.9306019  3.606429

Tuning parameter 'intercept' was held constant at a value of TRUE

Call:
lm(formula = .outcome ~ ., data = dat)

Residuals:
    Min      1Q  Median      3Q     Max
-43.101  -3.146  -0.132   3.173  17.735

Coefficients:
             Estimate Std. Error  t value Pr(>|t|)
(Intercept) 453.607920  10.778493   42.085  < 2e-16 ***
AT           -1.984670   0.016884 -117.544  < 2e-16 ***
V            -0.227863   0.008049  -28.308  < 2e-16 ***
AP            0.062865   0.010456    6.012 1.91e-09 ***
RH           -0.157308   0.004596  -34.231  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.504 on 7651 degrees of freedom
Multiple R-squared:  0.9305,    Adjusted R-squared:  0.9305
F-statistic: 2.561e+04 on 4 and 7651 DF,  p-value: < 2.2e-16
```



No conjunto de teste:

RMSE	Rsquared	MAE
4.769306	0.9214907	3.71388

Regressão Não-linear e KNN



Rodrigo A. Scarpel

rodrigo@ita.br

www.ief.ita.br/~rodrigo

Regressão Não Linear

- Para capturar relações não lineares entre as variáveis independentes e a variável dependente, pode-se utilizar:
    - Transformação dos dados (por exemplo, utilizando potência, logarítmo, raiz quadrada, entre outros) nas variáveis independents e/ou na variável dependente;
    - Modelos polinomiais: abordagem simples em que se emprega tranformações polinomiais de segunda ou de terceira ordem nas variáveis independentes;
    - Modelos Spline: ajuste de uma série de curvas suavizadas utilizando segmentos polinomiais (de terceira ordem) delimitados por nós;
    - Modelos aditivos generalizados (GAM): ajuste de modelos spline com nós selecionados automaticamente.
    - …

Transformações:

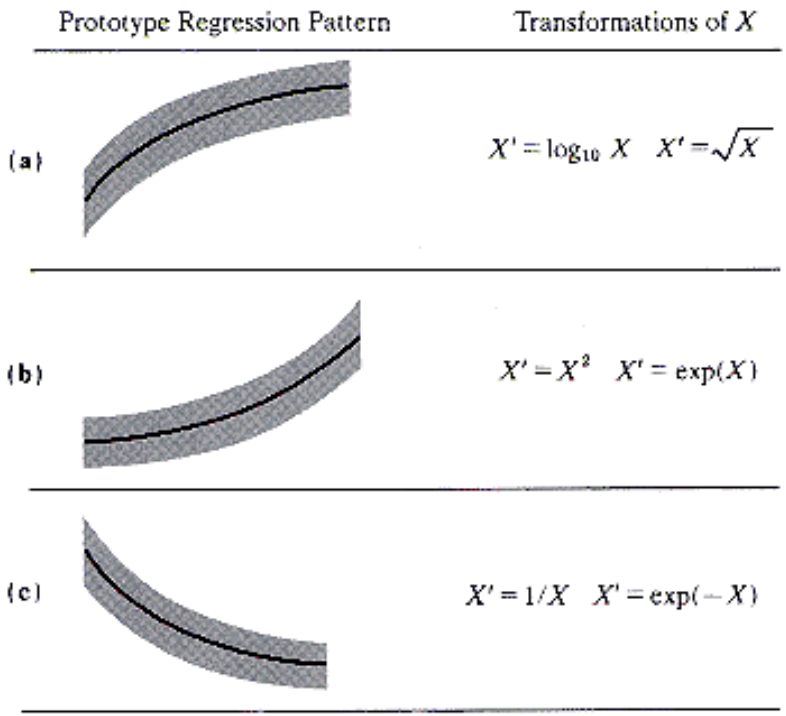
- A transformação das variáveis independentes (X), ou da variável dependente (Y), ou de ambas, normalmente, é suficiente para tornar um modelo apropriado (adequado).
- As transformações podem ser aplicadas para:
  - Ajustar para a linearização entre as variáveis independente e dependente (variância já é aproximadamente constante)
  - Satisfazer algumas hipóteses teóricas
  - Melhorar o ajuste e a qualidade das previsões geradas
  - Transformações comumente empregadas:

Function	Transformation	Linear Form
$Y = \alpha X^{\beta}$	$Y' = \log Y, X' = \log X$	$Y' = \log \alpha + \beta X'$
$Y = \alpha e^{\beta X}$	$Y' = \ln Y$	$Y' = \ln \alpha + \beta X$
$Y = \alpha + \beta \log X$	$X' = \log X$	$Y = \alpha + \beta X'$
$Y = \frac{X}{\alpha X - \beta}$	$Y' = \frac{1}{Y}, X' = \frac{1}{X}$	$Y' = \alpha - \beta X'$
$Y = \frac{e^{\alpha+\beta X}}{1+e^{\alpha+\beta X}}$	$Y' = \ln \frac{Y}{1-Y}$	$Y' = \alpha + \beta X$

Transformações:

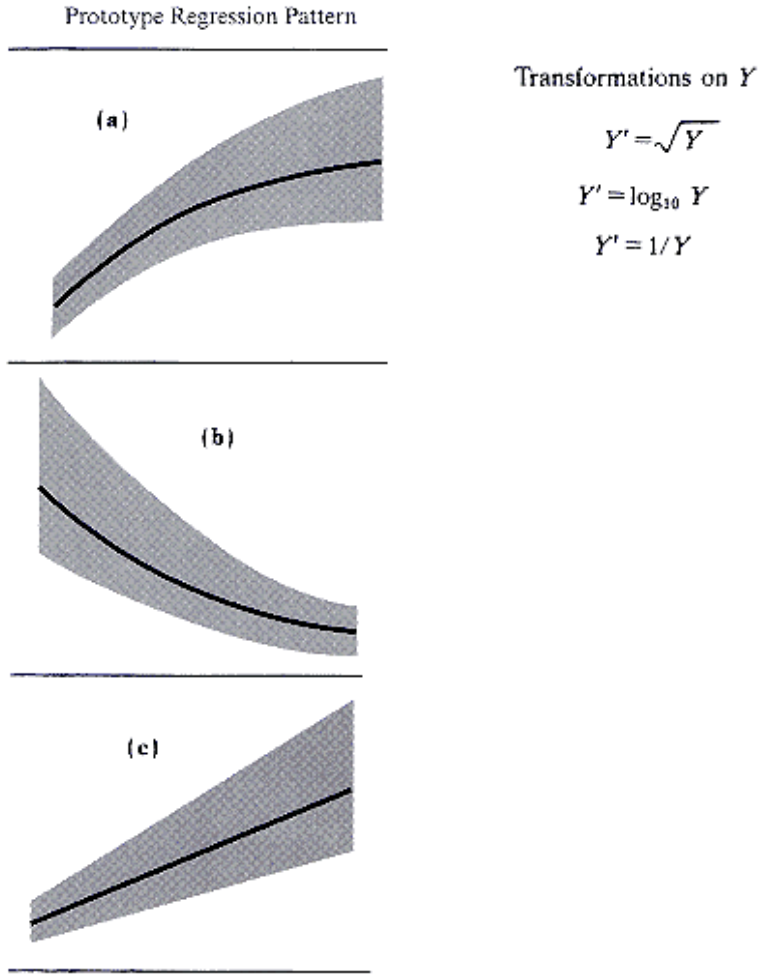
- Transformação das variáveis independentes (X):

Prototype Nonlinear Regression Patterns with Constant Error Variance and Simple Transformations of X.



WHEN ERROR VARIANCE IS CONSTANT, ONE NEEDS ONLY TRANSFORM X, NOT Y

- Transformação da variável dependente (Y):



Note: A simultaneous transformation on X may also be helpful or necessary.

Transformações:

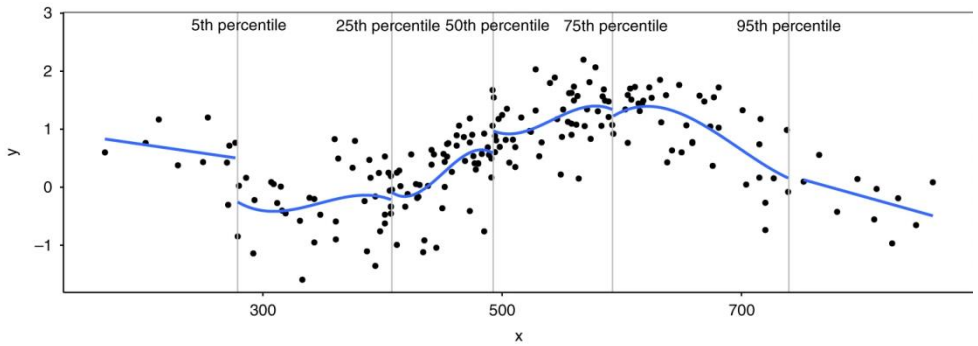
- Transformação das variáveis independentes:
  - Ex 1: modelo.log <- lm(y ~ log(x), data=treino)
  - Ex 2: modelo.raiz <- lm(y ~sqrt(x), data=treino)
- Modelos Polinomiais: no R há 2 formas de criar modelos polinomiais
  - Usando I( ). Ex: modelo.p2 <- lm(y ~ x + I(x^2), data=treino)
  - Usado a função poly. Ex: modelo.p3 <- lm(y ~ poly(x,3,raw=TRUE), data=treino)
- Utilizando os modelos:
  - Gerar previsões: previsao.nome <- modelo.nome %>% predict(teste)
  - Desempenho: RMSE(previsao.nome, teste$y); R2(previsao.nome, teste$y)

OBS: raw = FALSE, poly() computes an orthogonal polynomial

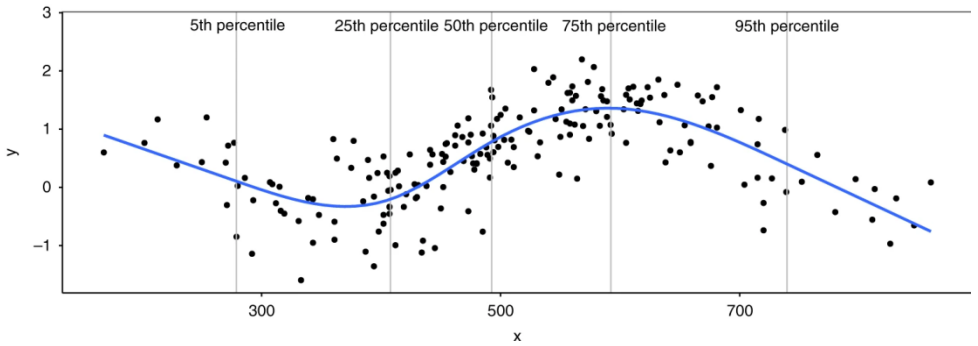
Regressão Não Linear

- Regressão Spline: permite a interpolação suave de polinômios (em geral de terceira ordem) entre pontos fixos, chamados de nós. Assim, os splines são uma série de segmentos de polinômios unidos por nós.

6 polinômios cúbicos:



Splines (5 nós):



- No R:

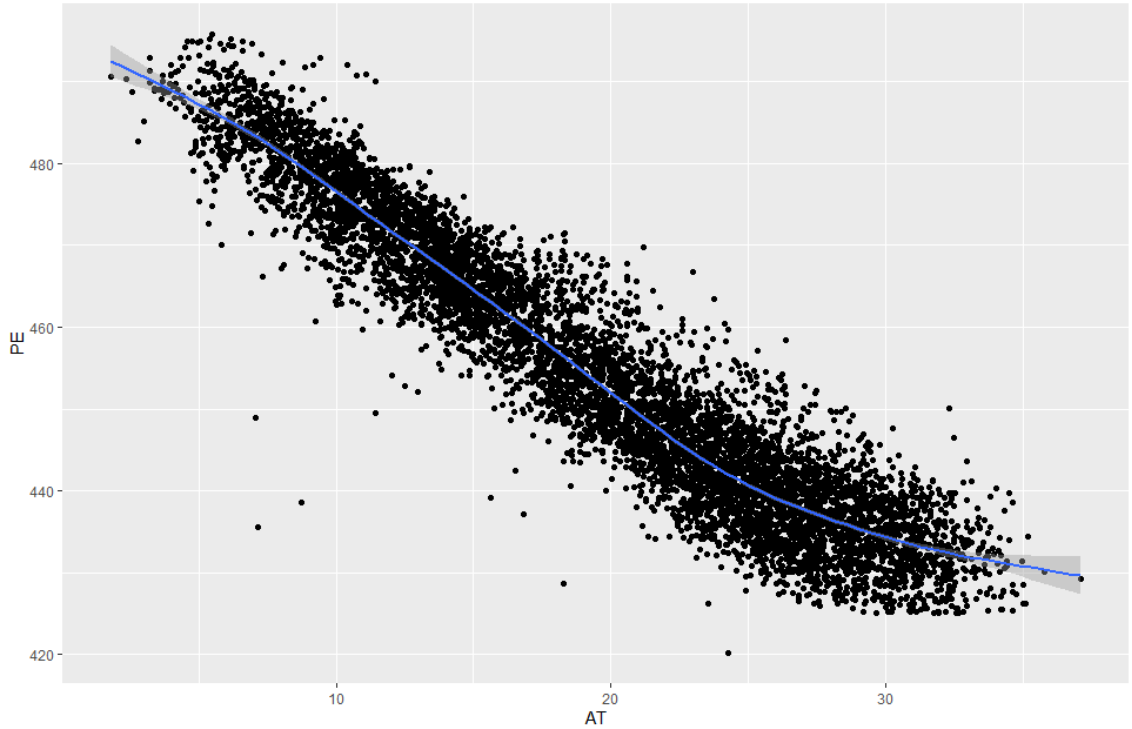
  - Ex 1: knots <- quantile(treino$x, p = c(0.05,0.25, 0.5, 0.75,0.95))
    modelo <- lm(y ~ bs(x, knots = knots), data = treino)

  - Ex 2: modelo <- lm(y ~ bs(x, df = 5), data = treino) # p = 20^{th},40^{th},60^{th},80^{th}

Regressão Não Linear

- Regressão GAM: técnica para criar, automaticamente, regressões spline (não há a necessidade de indicar onde estão os nós e/ou quantos splines serão feitos).

- No R: modelo <- gam(y ~ s(x), data = treino)



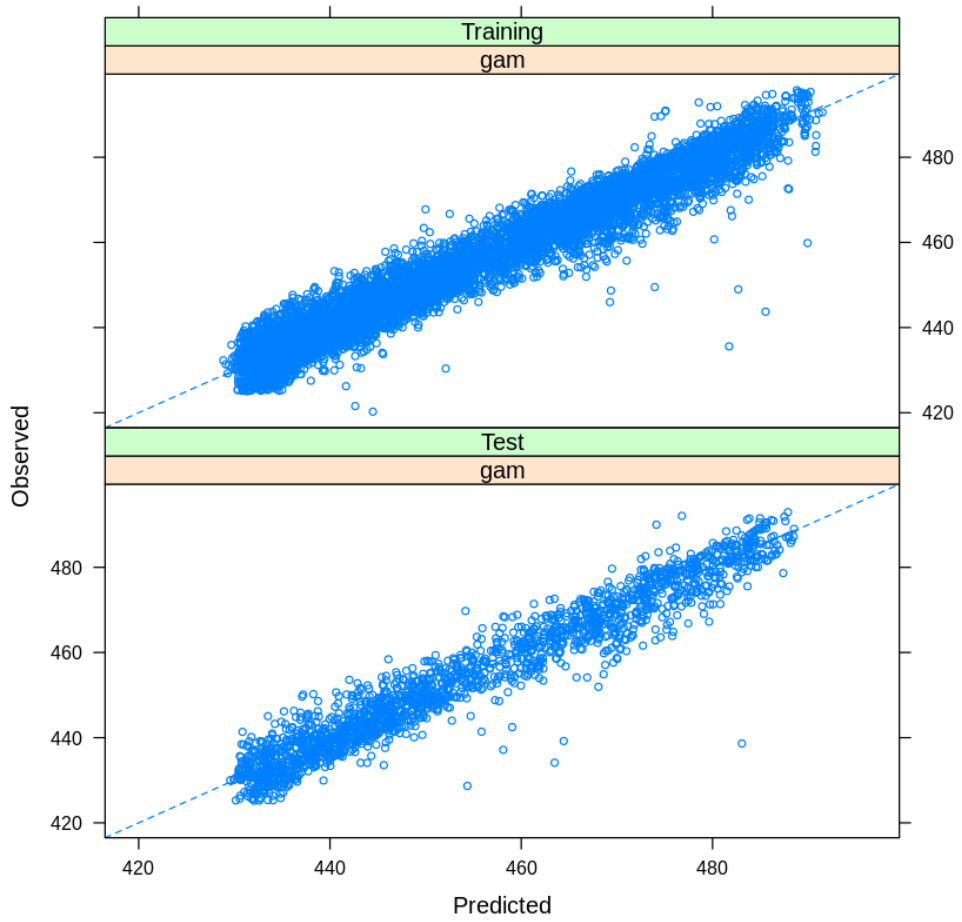

```
ggplot(treino, aes(AT,PE) ) +
  geom_point() +
  stat_smooth(method = 'gam',
              formula = y ~ s(x))
```

Regressão Não Linear

- Regressão GAM: Previsão de Produção de Energia

Resultado:

```
Parametric coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 454.39135    0.04684    9701   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Approximate significance of smooth terms:
       edf Ref.df       F p-value
s(V)  8.161      9  154.98  <2e-16 ***
s(AP) 7.662      9   33.82  <2e-16 ***
s(AT) 7.024      9 1504.73  <2e-16 ***
s(RH) 5.688      9   74.20  <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

R-sq.(adj) =  0.942   Deviance explained = 94.3%
GCV = 16.861  Scale est. = 16.796    n = 7656
```



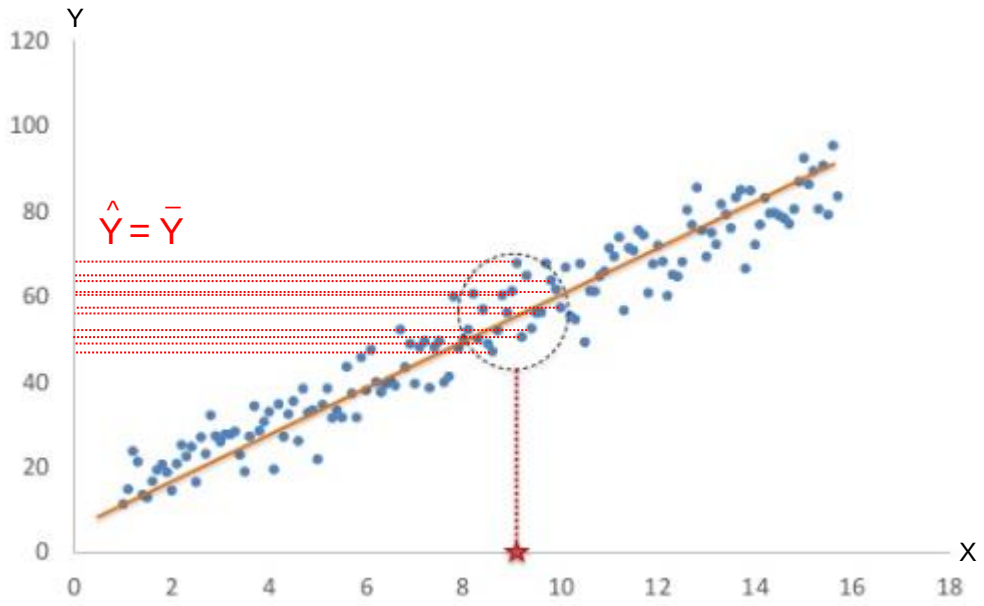
No conjunto de teste:

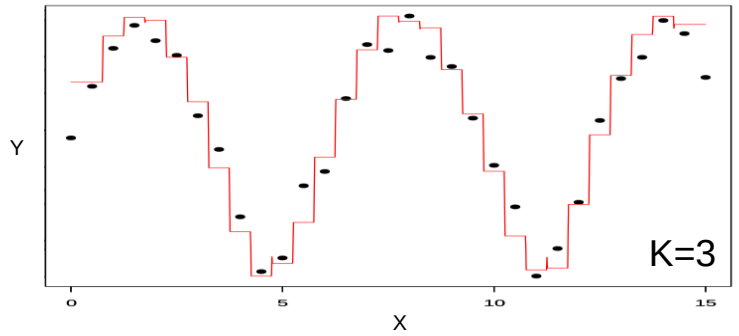
RMSE	Rsquared	MAE
4.37934	0.9337901	3.351107

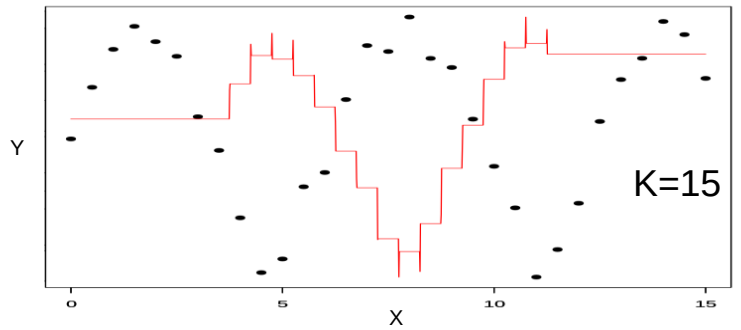
Regressão KNN

K-ésimo vizinho mais próximo:

Para minimizar o erro de previsão de uma observação, toma-se os k vizinhos mais próximos (distância Euclideana) e atribui-se à observação a média dos valores da variável dependente (Y)*:*



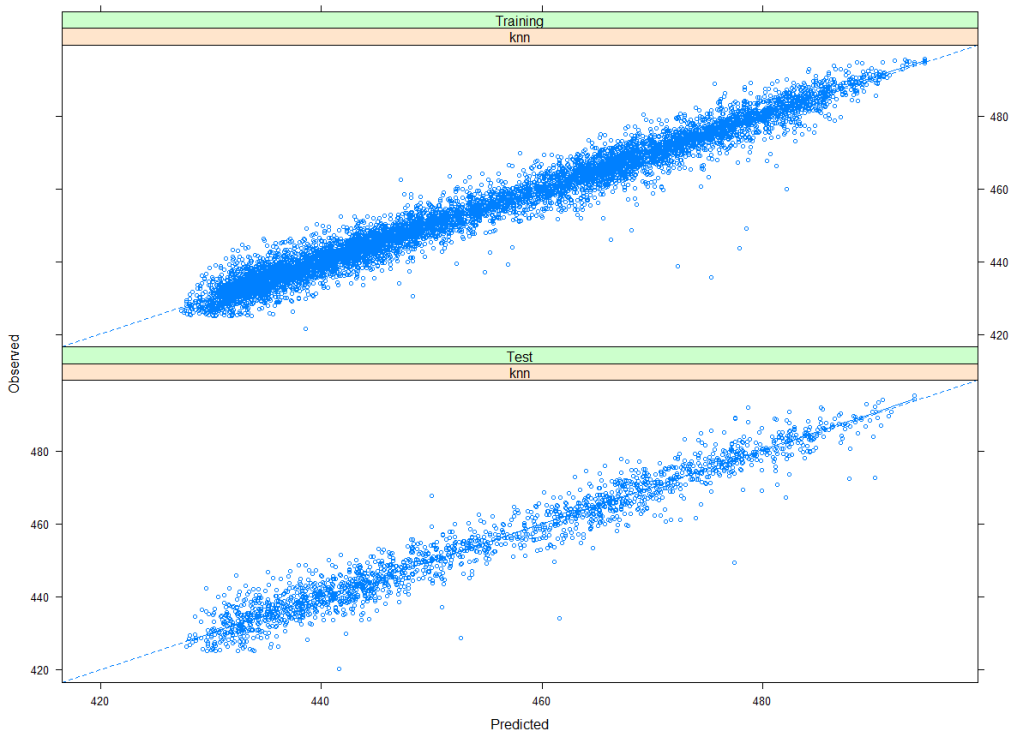




Regressão KNN

- Exemplo: Previsão de Produção de Energia

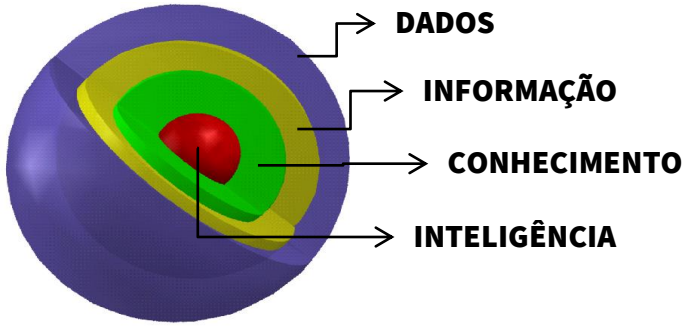
```
k-Nearest Neighbors

7656 samples
   4 predictor

Pre-processing: centered (4), scaled (4)
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 6889, 6892, 6889, 6890, 6891, 6890, ...
Resampling results across tuning parameters:

  k   RMSE      Rsquared   MAE
  1   4.382565  0.9355370  2.970644
  2   4.027577  0.9446565  2.827607
  3   3.897432  0.9478720  2.770849
  4   3.860072  0.9488077  2.768973
  5   3.862431  0.9486981  2.796808
  6   3.858485  0.9487799  2.813904
  7   3.883349  0.9481238  2.851669
  8   3.894265  0.9478418  2.876094
  9   3.905729  0.9475296  2.897263
 10   3.915136  0.9472704  2.910032
 11   3.931469  0.9468486  2.932671
 12   3.951573  0.9462983  2.955901
 13   3.970147  0.9457832  2.973805
 14   3.981043  0.9454858  2.987337
 15   3.990032  0.9452461  2.998126
 16   4.002209  0.9449296  3.011055
 17   4.013543  0.9446237  3.023955
 18   4.023173  0.9443738  3.034552
 19   4.037096  0.9439910  3.047788
 20   4.050442  0.9436156  3.061391
 21   4.064005  0.9432447  3.075035
 22   4.075623  0.9429300  3.088976
 23   4.088646  0.9425720  3.101926
 24   4.100154  0.9422447  3.113049
 25   4.110943  0.9419541  3.123947
 26   4.122921  0.9416206  3.132816
 27   4.131056  0.9413880  3.141813
 28   4.138187  0.9411904  3.148159
 29   4.145094  0.9409986  3.155950
 30   4.149672  0.9408744  3.161866

RMSE was used to select the optimal model using the smallest value.
The final value used for the model was k = 6.
```



No conjunto de teste:

```
     RMSE  Rsquared       MAE
3.8226652 0.9501755 2.7917206
```

SVR, CART e Ensembles em Regressão



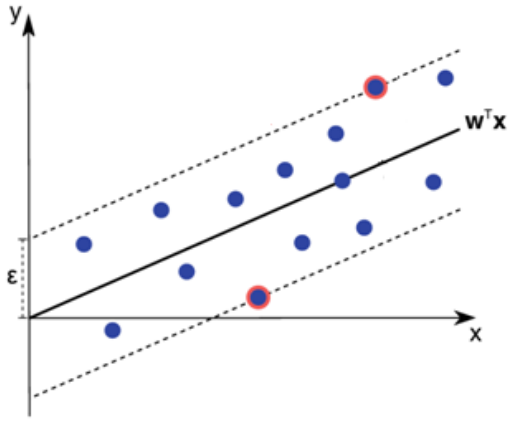
Rodrigo A. Scarpel

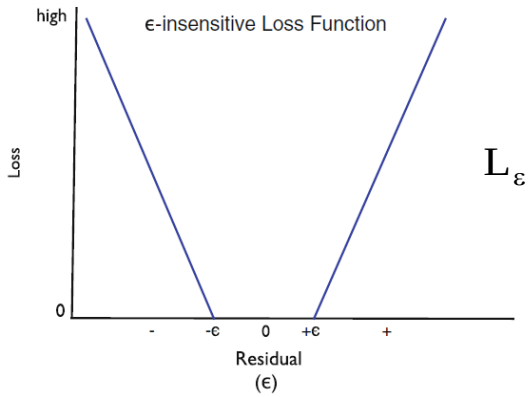
rodrigo@ita.br

www.ief.ita.br/~rodrigo

Support Vector Regression

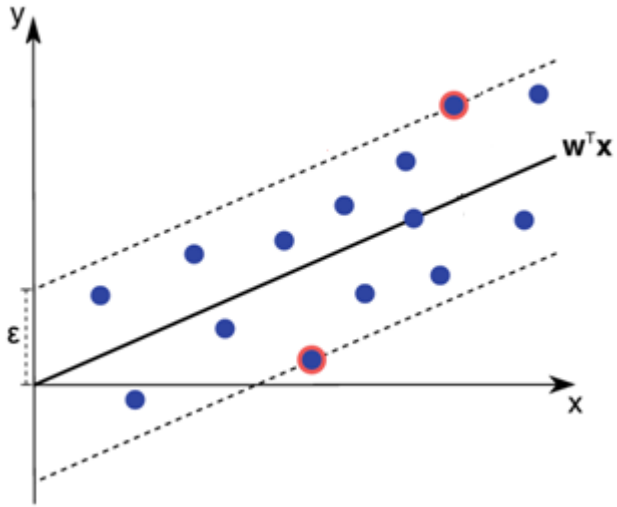
- Funções de perda (comumente usadas):
  - SE (erro quadrático): $L(y,f(x)) = (y - f(x))^2$
  - AE (erro absoluto): $L(y,f(x)) = |y - f(x)|$
  - Huber: $$L(y,f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2, & \text{se } |y - f(x)| < \mu \\ \mu|y - f(x)| - \frac{\mu^2}{2}, & \text{caso contrário} \end{cases}$$
- SVR:

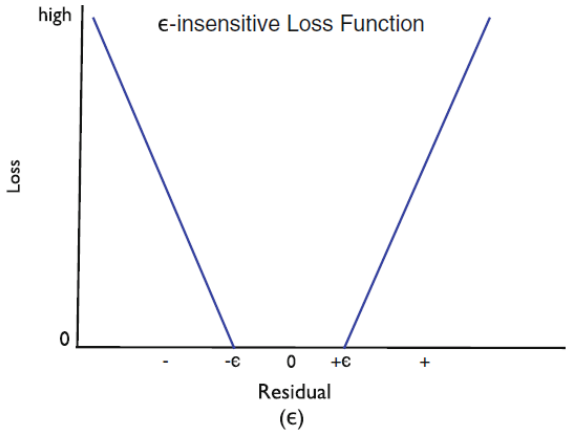




$$L_\varepsilon(y,f(x)) = \begin{cases} 0, & \text{se } |y - f(x)| \le \varepsilon \\ |y - f(x)| - \varepsilon, & \text{caso contrário} \end{cases}$$

Support Vector Regression

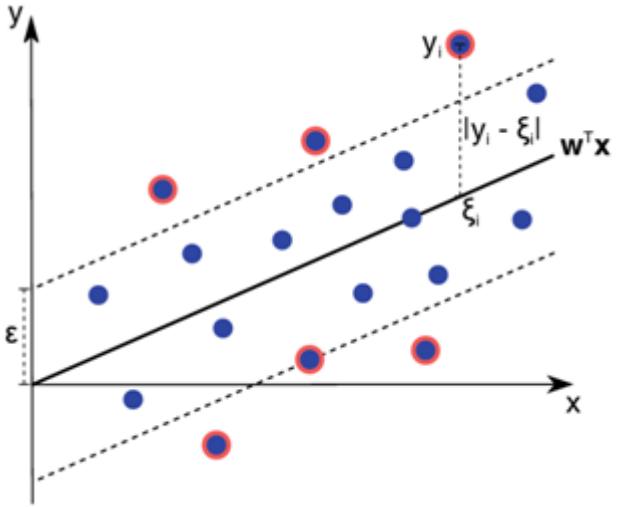




$$\min \frac{1}{2} \| \mathbf{w} \|^2$$

$$s.t.\ y_i - \mathbf{w}_1 \cdot \mathbf{x}_i - b \le \varepsilon$$

$$\mathbf{w}_1 \cdot \mathbf{x}_i + b - y_i \le \varepsilon$$



$$\text{Min} \quad \frac{1}{2} \| \mathbf{w} \|^2 + C \sum_{i=1}^{m} (\xi_i + \xi_i^*)$$

$$y_i - (\mathbf{w} \cdot \mathbf{x}_i) - b \le \varepsilon + \xi_i$$

$$(\mathbf{w} \cdot \mathbf{x}_i) + b - y_i \le \varepsilon + \xi_i^*$$

$$\xi_i\ , \xi_i^* \ge 0, i = 1, \ldots, m$$

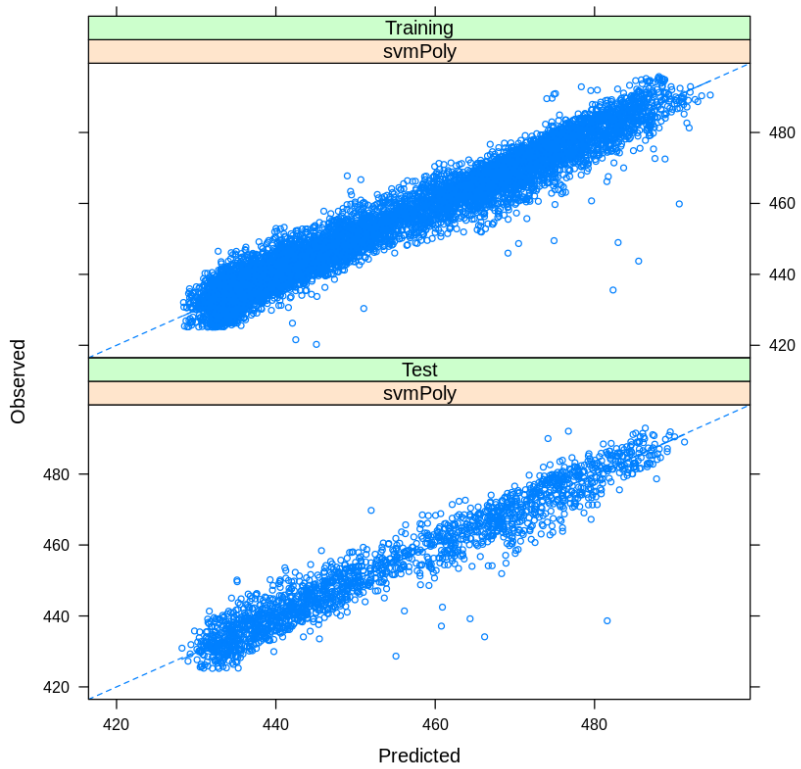
em que

C é uma constante de penalização (C>0)

Support Vector Regression

- Exemplo: Previsão de Produção de Energia (Caso Polinomial)

```
Support Vector Machines with Polynomial Kernel

7656 samples
   4 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 6890, 6890, 6891, 6890, 6891, 6889, ...
Resampling results across tuning parameters:

  degree  scale  C     RMSE      Rsquared   MAE
  1       0.001  0.25  5.414354  0.9104752  4.327534
  1       0.001  0.50  4.919687  0.9199796  3.943724
  1       0.001  1.00  4.657685  0.9263476  3.735225
  1       0.001  2.00  4.554549  0.9291527  3.645879
  1       0.010  0.25  4.539153  0.9296200  3.630480
  ⋮                    ⋮                    ⋮
  3       0.100  0.50  4.129280  0.9418801  3.203991
  3       0.100  1.00  4.126427  0.9419683  3.200136
  3       0.100  2.00  4.125010  0.9420107  3.197748
  3       1.000  0.25  4.127047  0.9419369  3.195986
  3       1.000  0.50  4.127114  0.9419366  3.196076
  3       1.000  1.00  4.127083  0.9419375  3.196099
  3       1.000  2.00  4.128687  0.9419257  3.198146

RMSE was used to select the optimal model using the smallest value.
The final values used for the model were degree = 3, scale = 0.1 and C = 2.
```

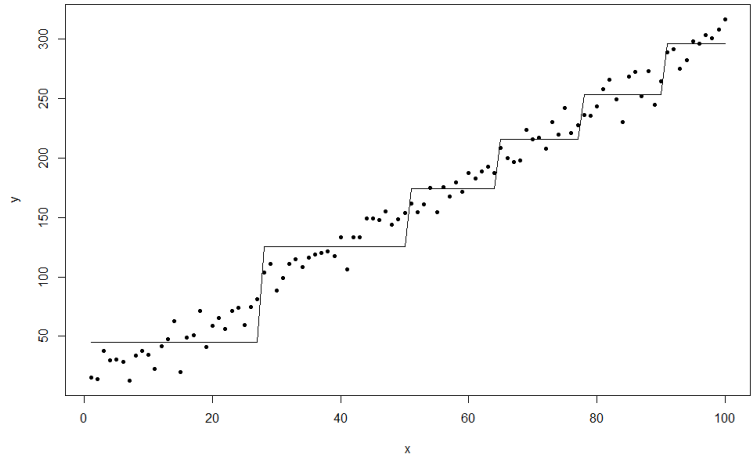


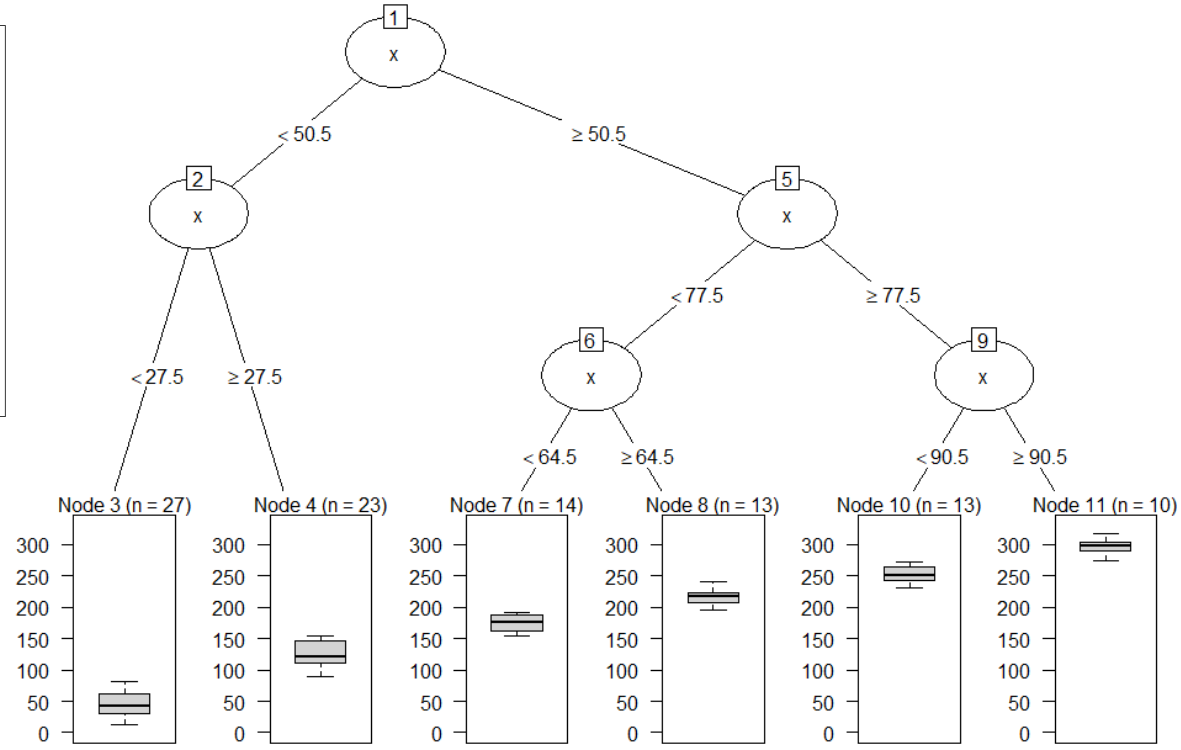
No conjunto de teste:

RMSE	Rsquared	MAE
4.395156	0.9334925	3.332211

Regressão CART / AID

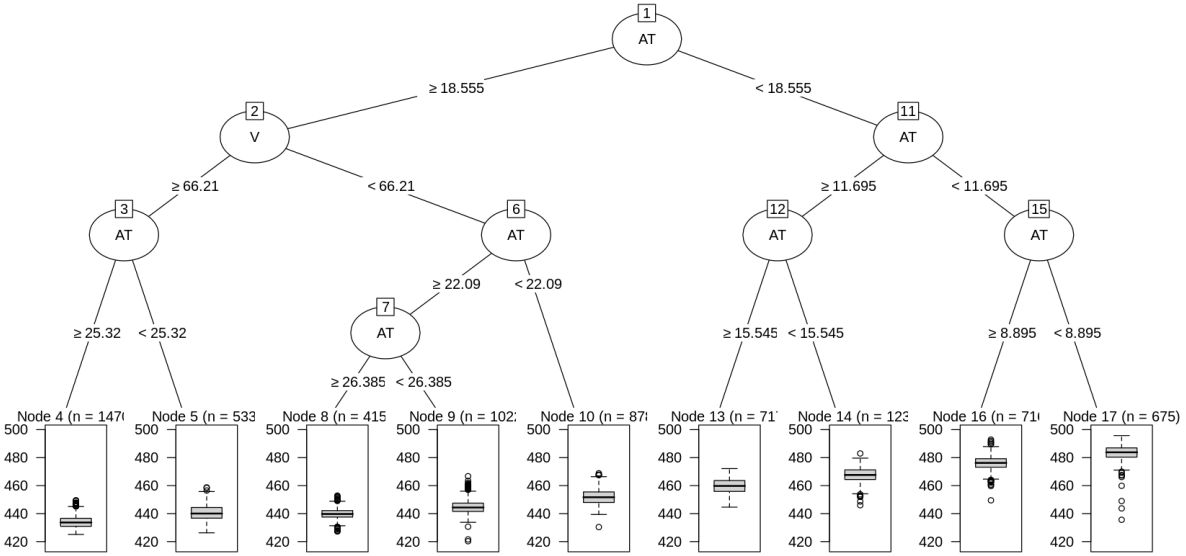
- Sistema de regressão que particiona o espaço de atributos de forma a criar regras para definir as previsões
- Resultado: um conjunto de regras e uma árvore (diagrama)

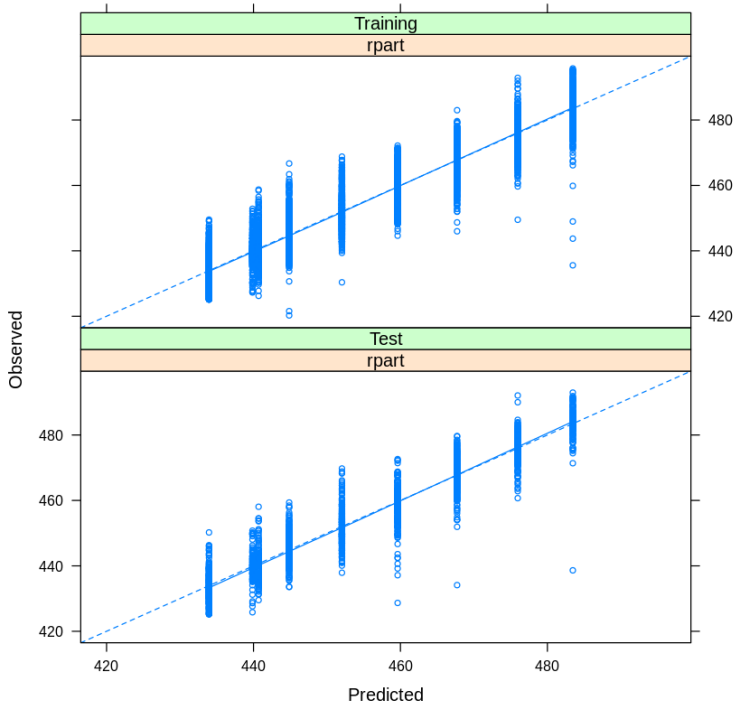




Regressão CART / AID

- Exemplo: Previsão de Produção de Energia






```
CART

7656 samples
   4 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 6890, 6891, 6891, 6889, 6892, 6889, ...
Resampling results across tuning parameters:

  cp           RMSE       Rsquared   MAE
  0.003222235   5.046109  0.9127141   3.950501
  0.003297068   5.070537  0.9118699   3.973783
  0.007977615   5.317218  0.9030559   4.156606
  0.008743808   5.493585  0.8965519   4.293378
  0.013277272   5.948714  0.8787278   4.653023
  0.018153683   6.137882  0.8708569   4.814033
  0.057575763   6.915921  0.8351614   5.501356
  0.080338750   8.647896  0.7427616   7.016755
  0.724611179  13.122185  0.7111319  11.164022

RMSE was used to select the optimal model using the smallest value.
The final value used for the model was cp = 0.003222235.
```

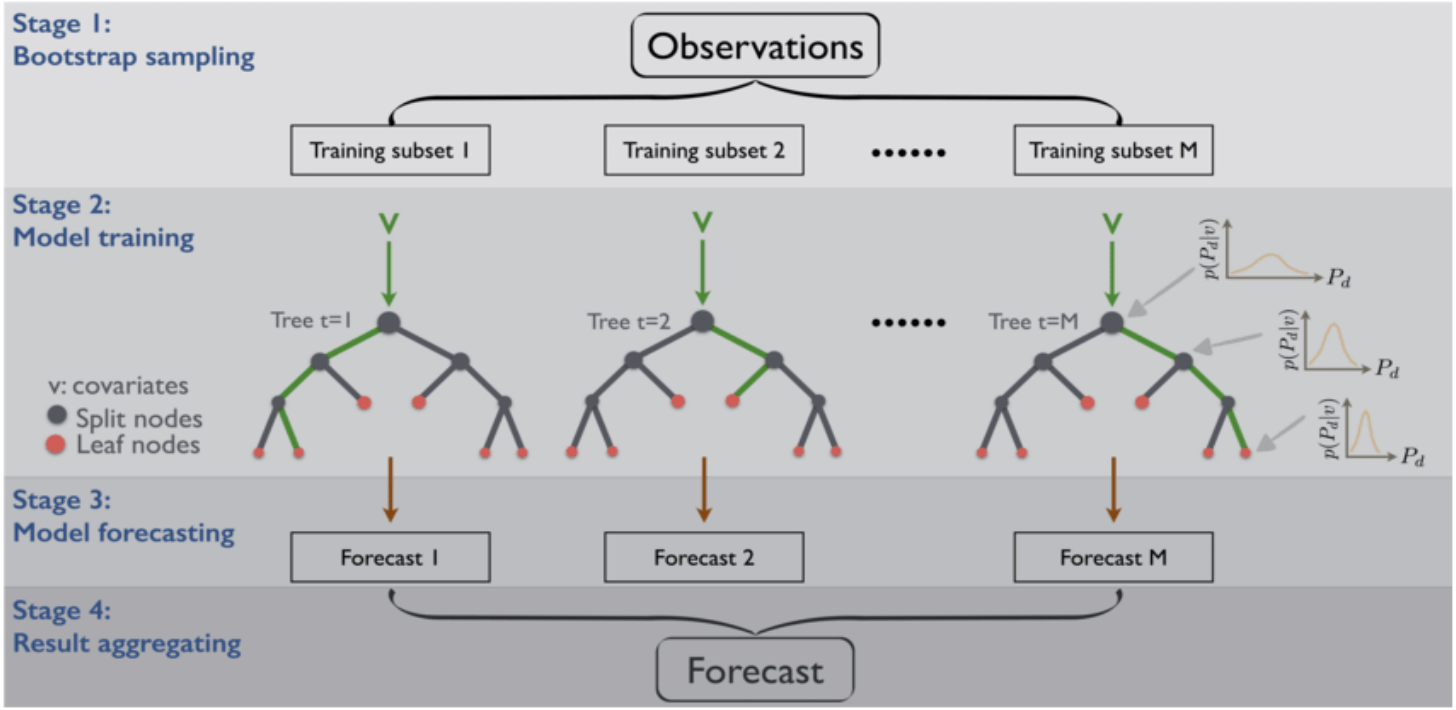
No conjunto de teste:

RMSE	Rsquared	MAE
5.257559	0.9046005	4.01715

Métodos de Ensemble

- **Bagging (Bootstrapp AGGregatING):**

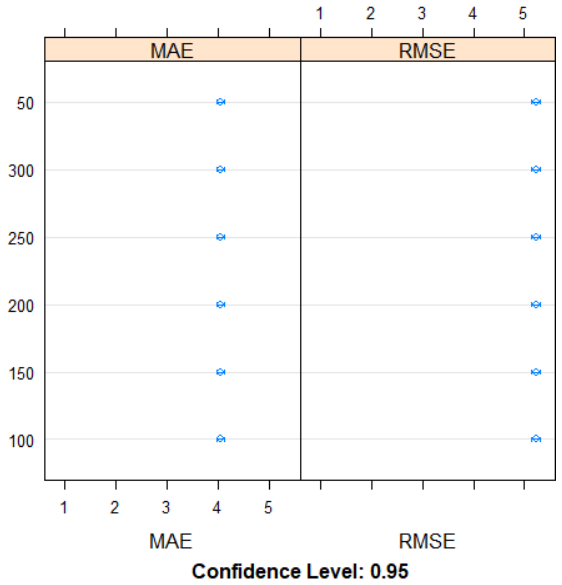
  No estágio 1, cria-se amostras aleatórias de conjuntos de treinamento (com reposição), no estágio 2 cria-se um classificador (CART) para cada conjunto de treinamento e nos estágios 3 e 4 combina-se a previsão dos modelos obtidos (utilizando a média dos resultados).



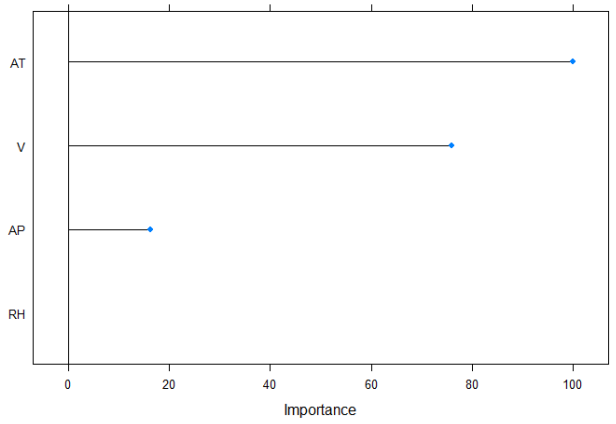
Regressão com Bagging

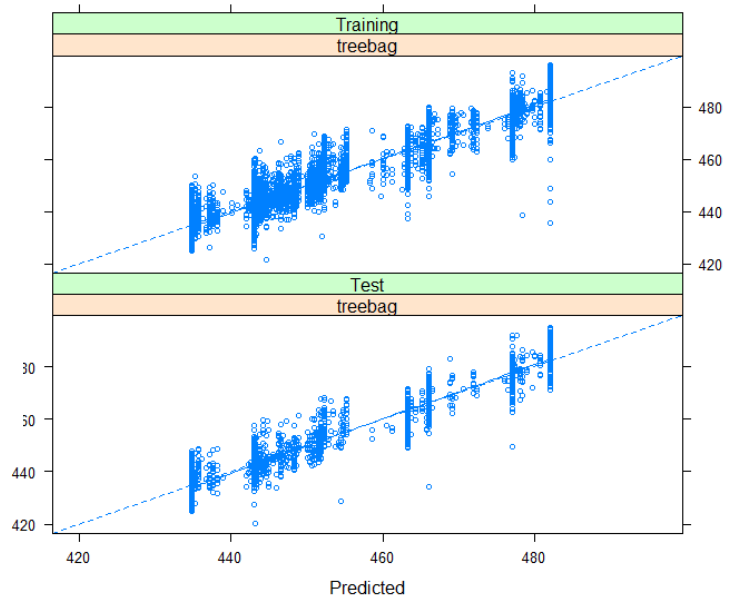
- Exemplo: Previsão de Produção de Energia

Número ideal de árvores:



Importância relativa das variáveis:





Resultado com 100 árvores:

```
Bagged CART

7656 samples
   4 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 6890, 6890, 6889, 6889, 6892, 6891, ...
Resampling results:

  RMSE     Rsquared   MAE
  5.19298  0.9079767  4.053402
```

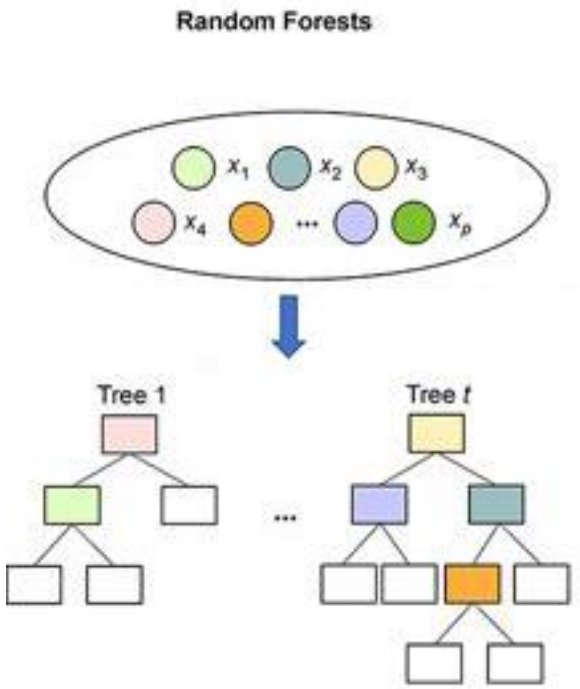
No conjunto de teste:

RMSE	Rsquared	MAE
5.24988	0.905395	4.012066

Métodos de Ensemble

- **Random Forest:**

A Random Forest faz uso do mesmo método do Bagging, ou seja, no estágio 1 cria-se amostras aleatórias de conjuntos de treinamento (com reposição), no estágio 2 cria-se uma regressão (CART) para cada conjunto de treinamento e nos estágios 3 e 4 combina-se a previsão dos modelos obtidos (utilizando a média dos resultados).

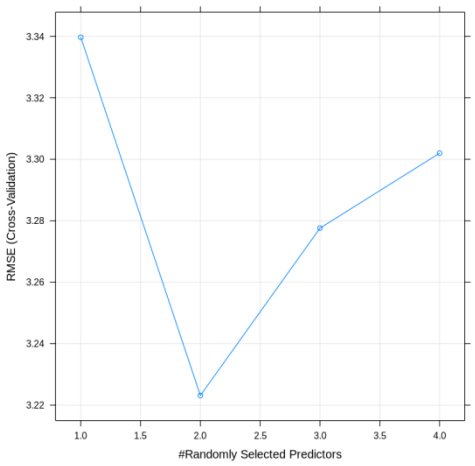


A diferença é que em cada classificador (CART) um subconjunto das variáveis independentes candidatas para compor a regressão são aleatóriamente escolhidas (há um número máximo de variáveis default).

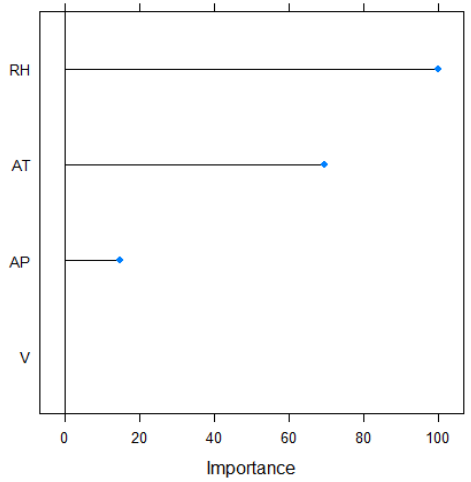
Regressão com Random Forest

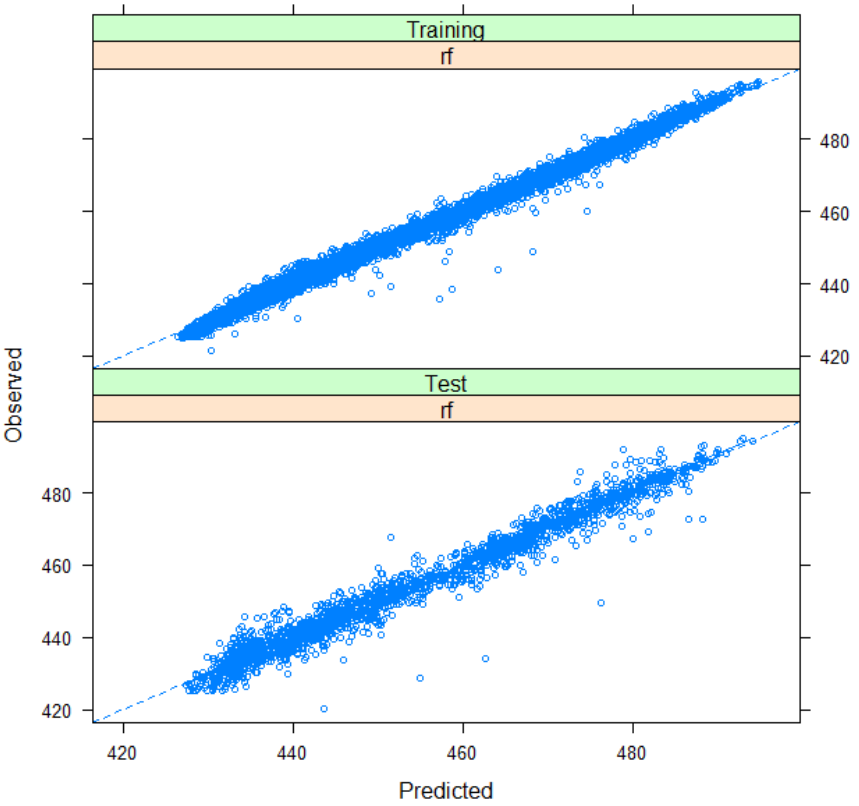
- Exemplo: Previsão de Produção de Energia

Número ideal variáveis por árvore:



Importância relativa das variáveis:





Resultado com 300 árvores (2 variáveis por árvore):

```
Random Forest

7656 samples
   4 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 6890, 6890, 6889, 6889, 6892, 6891, ...
Resampling results:

  RMSE     Rsquared   MAE
  3.22829  0.9641018  2.324253

Tuning parameter 'mtry' was held constant at a value of 2
```

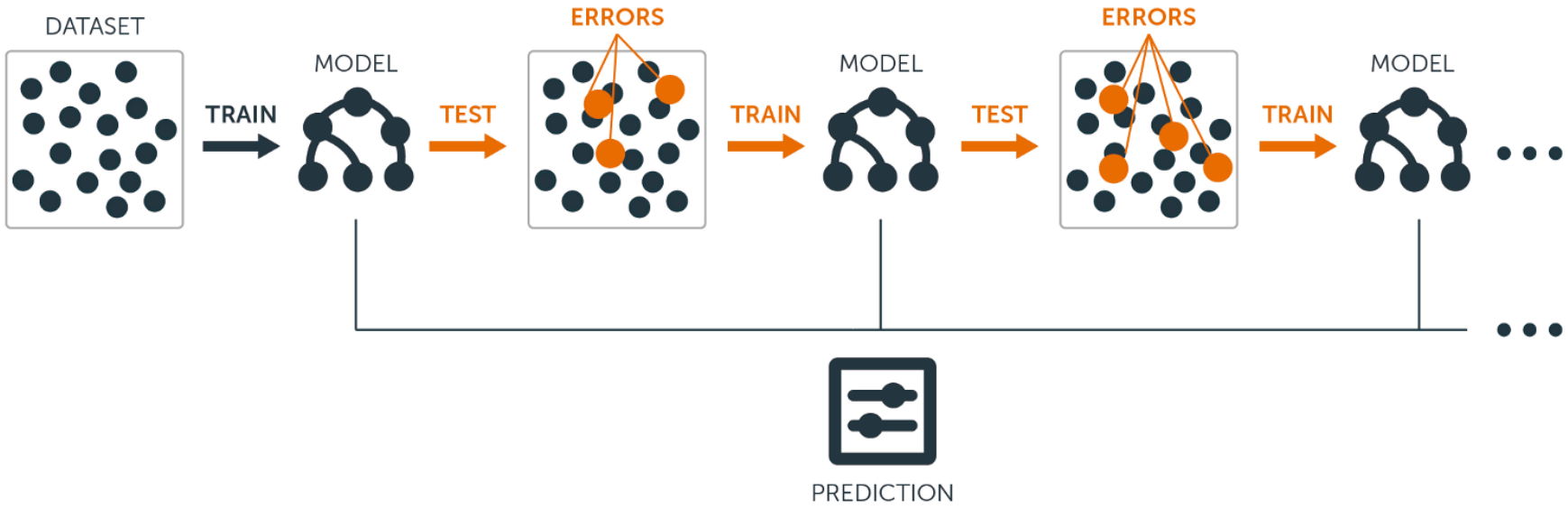
No conjunto de teste:

RMSE	Rsquared	MAE
3.385121	0.9605958	2.350088

Métodos de Ensemble

- **Gradiente Boosting (GB):**

  O GB é uma técnica de treinamento sequencial em que o primeiro modelo é criado para prever a variável resposta e os modelos subsequentes são criados visando prever os erros de previsão.



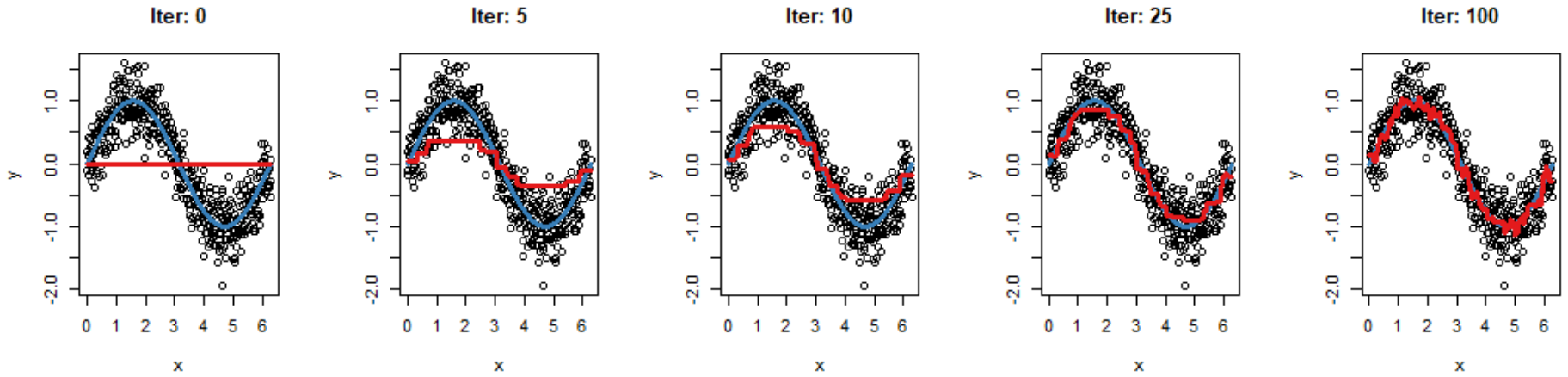
OBS: O GB auxilia na redução do erro de previsão (mas tende a gerar overfitting).

Métodos de Ensemble

- **Gradiente Boosting (GB):**

Treinamento sequencial de acordo com os erros – algoritmo básico:

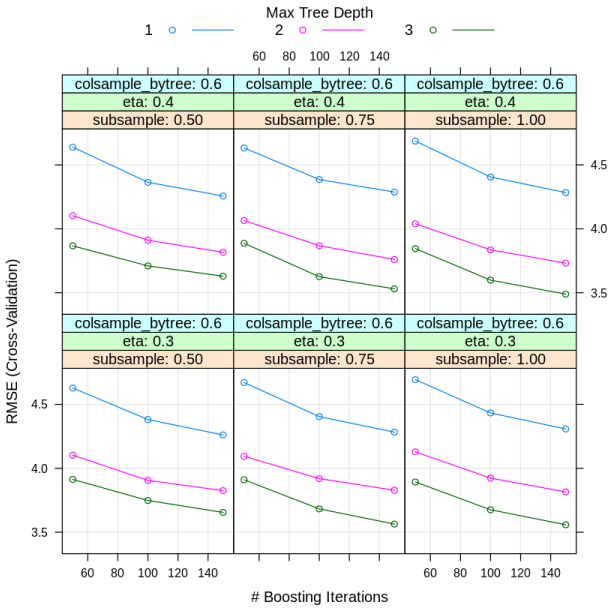
1. Ajuste um CART aos dados: $F_1(x) = y$,
2. Depois ajuste o próximo CART aos resíduos do CART anterior: $h_1(x) = y - F_1(x)$,
3. Adicione o novo CART ao algoritmo: $F_2(x) = F_1(x) + h_1(x)$,
4. Ajuste o próximo CART aos resíduos de F2: $h_2(x) = y - F_2(x)$,
5. Adicione o novo CART ao algoritmo: $F_3(x) = F_2(x) + h_2(x)$,
6. Continue o procedimento até algum mecanismo (ex. validação cruzada) indicar para parar.

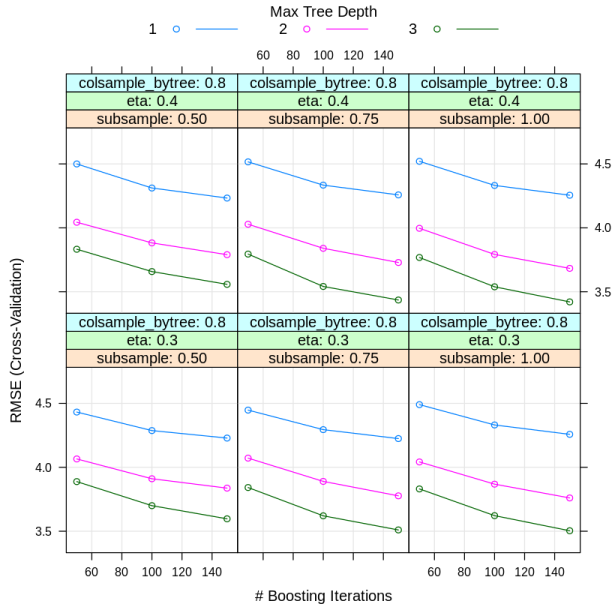


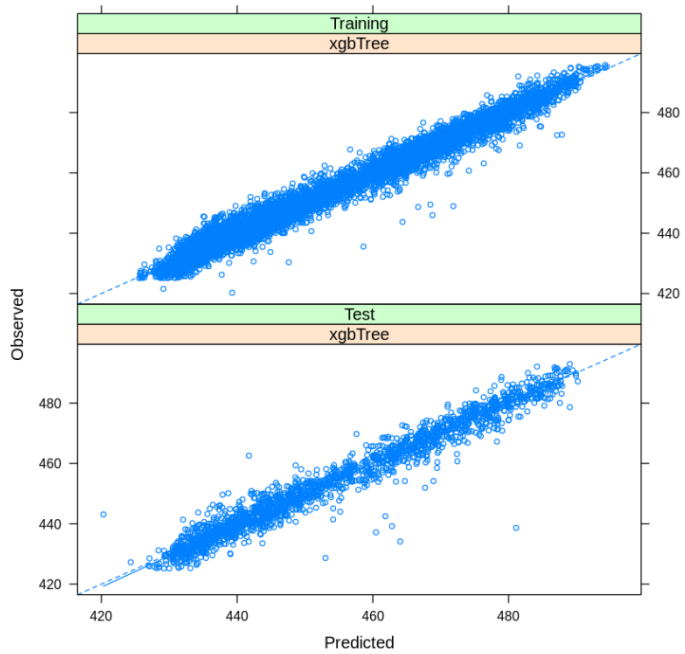
Regressão com Gradiente Boosting

- Exemplo: Previsão de Produção de Energia

Valor ideal para os parâmetros:







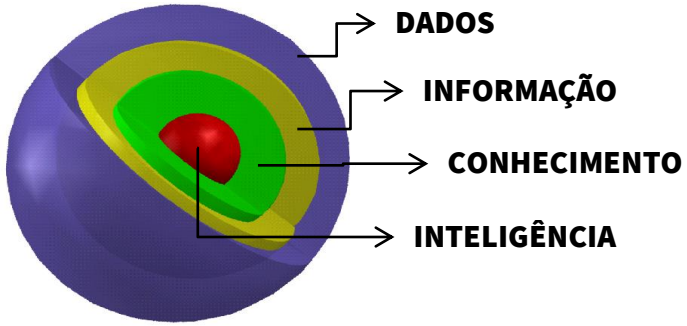

```
Acurácia na validação cruzada:
RMSE: 3.42
RSquared: 0.9598
```

```
Importância das variáveis:
xgbTree variable importance
   Overall
V  100.000
AT  90.304
AP   1.693
RH   0.000
```

No conjunto de teste:

RMSE	Rsquared	MAE
3.593413	0.9554286	2.518827

Métodos Prescritivos (Prescriptive Analytics)



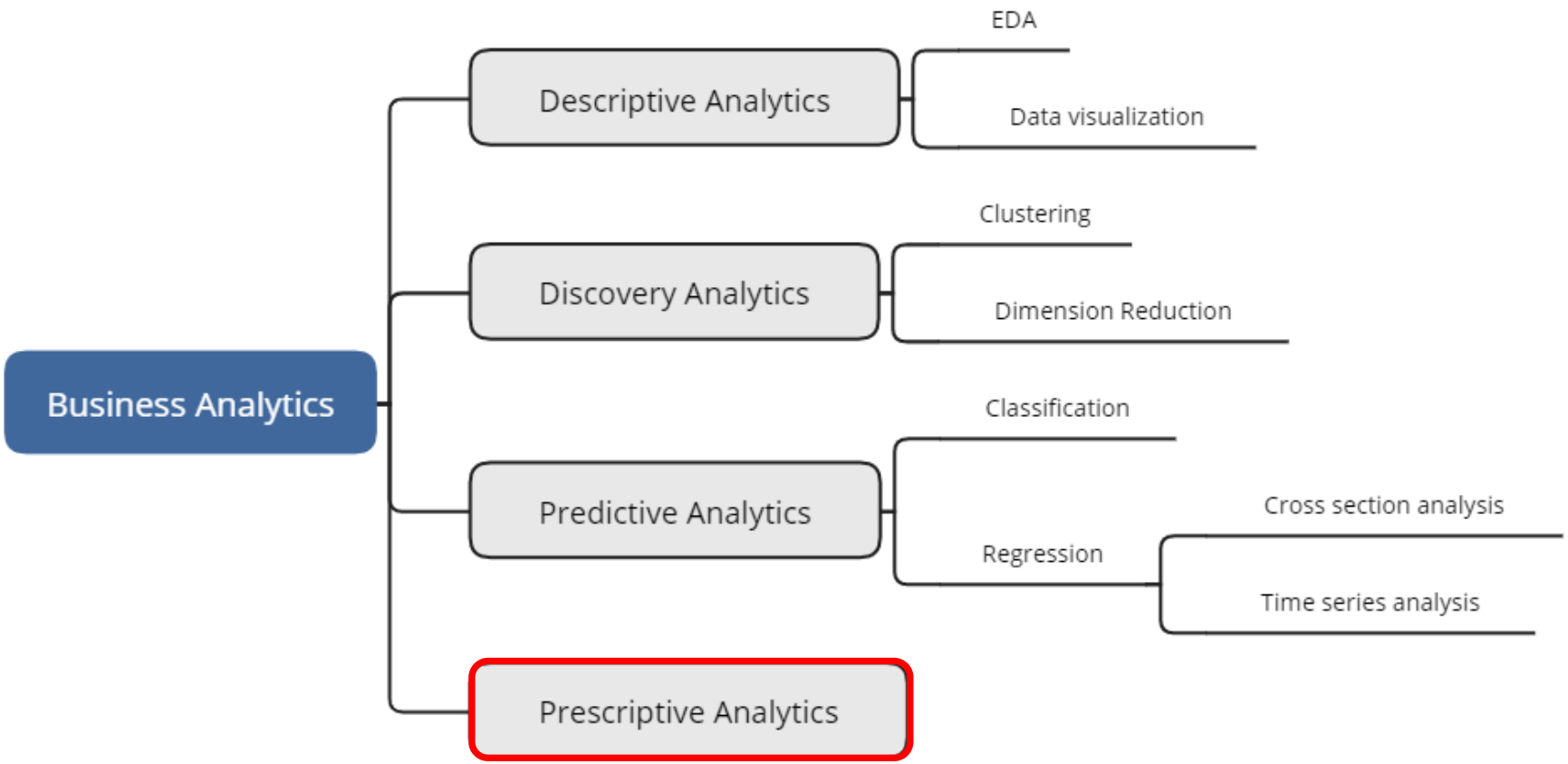
Rodrigo A. Scarpel

rodrigo@ita.br

www.ief.ita.br/~rodrigo

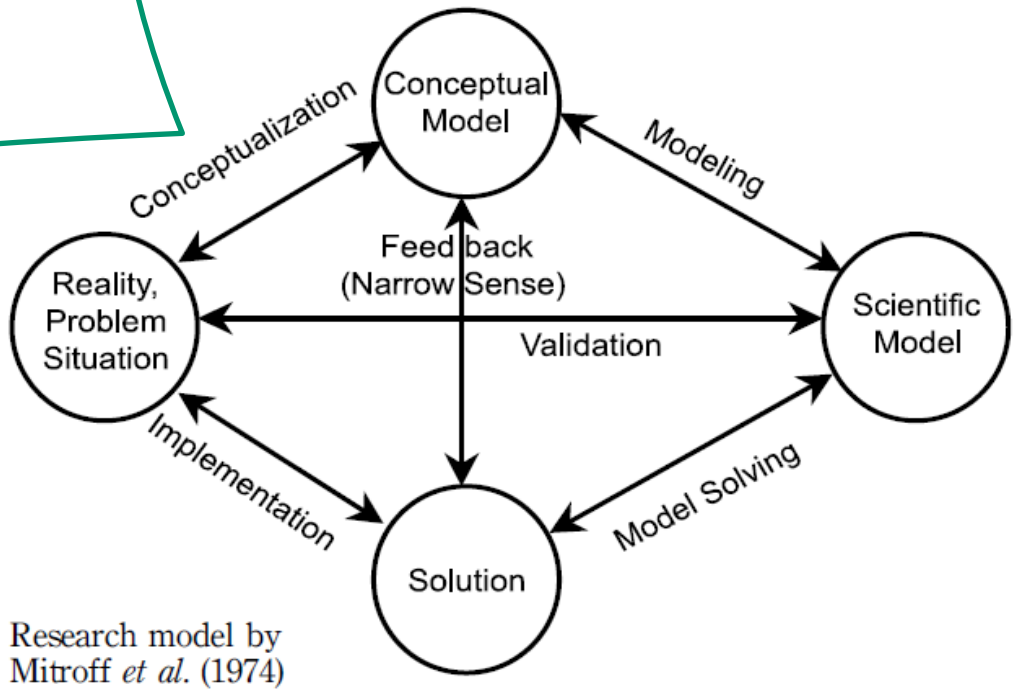
Introdução:

Prescriptive Analytics: the "final frontier of analytic capabilities. It entails the application of mathematical and computational sciences and suggests decision options to take advantage of the results of descriptive and predictive analytics.



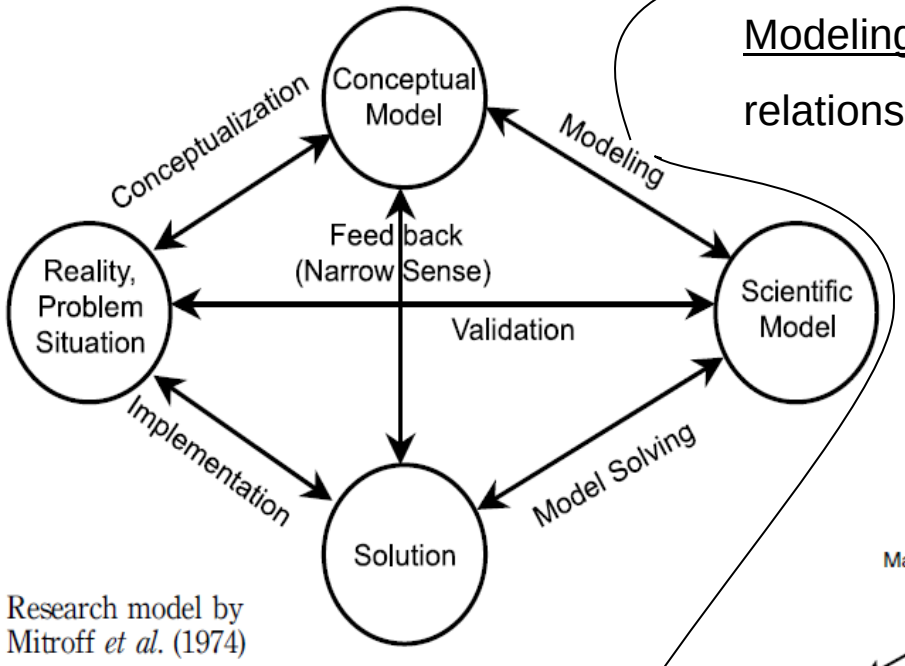
Modelagem e simulação:

Conceptualization: decisions about the variables that need to be included in the model, and the scope of the problem and model to be addresses.



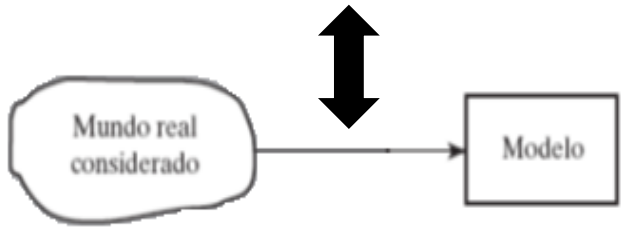
Research model by Mitroff *et al.* (1974)

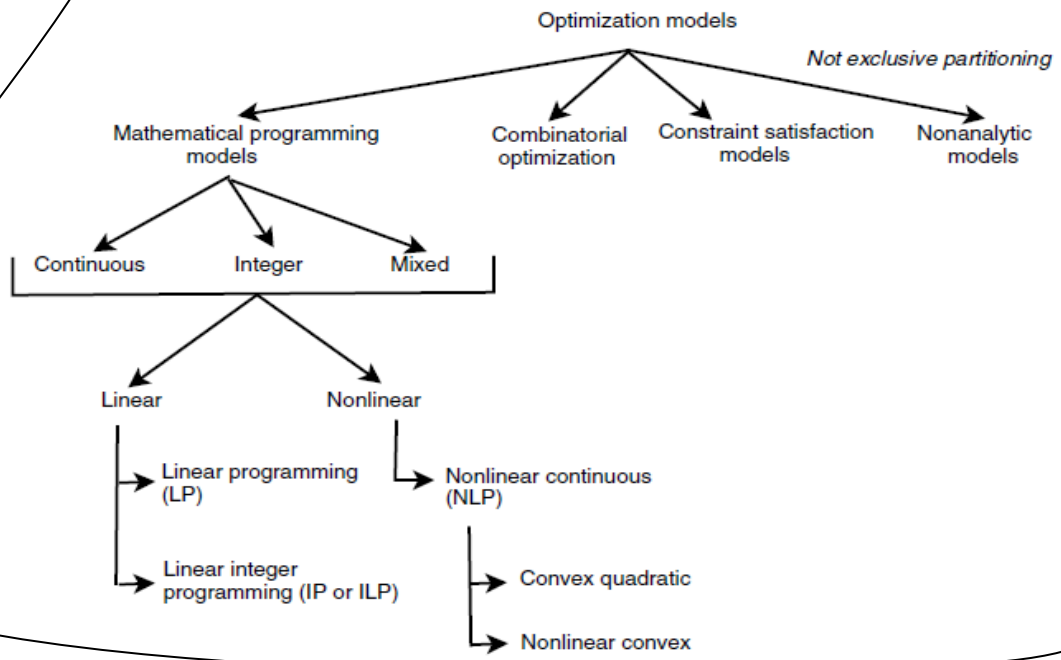
Modelagem e simulação:



Research model by Mitroff *et al.* (1974)

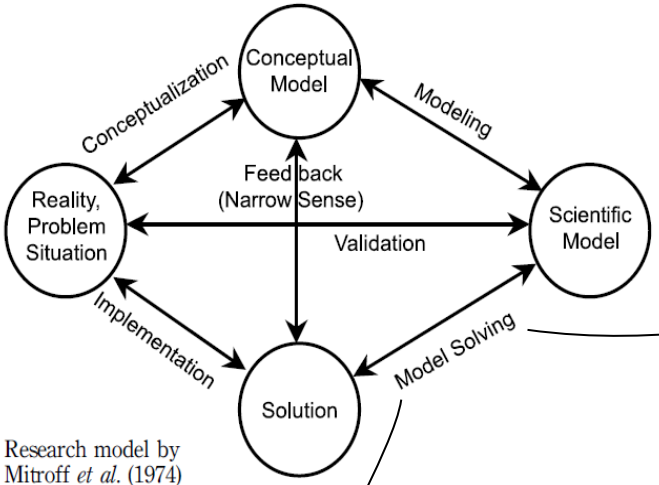
Modeling: build the quantitative model by defining causal relationships between the variables



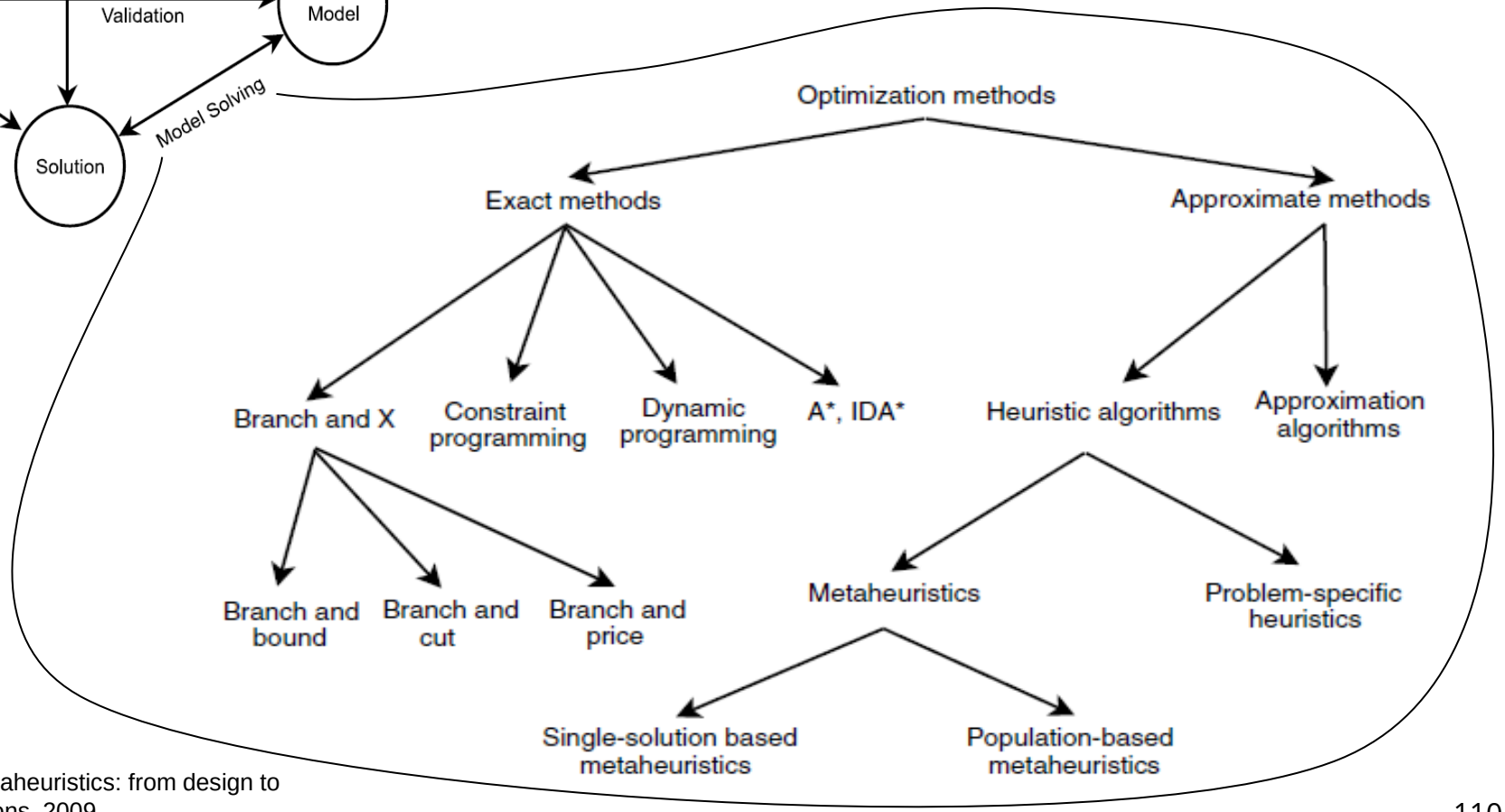


Fonte: TALBI, EL-GHAZALI. Metaheuristics: from design to implementation. John Wiley & Sons, 2009.

Modelagem e simulação:

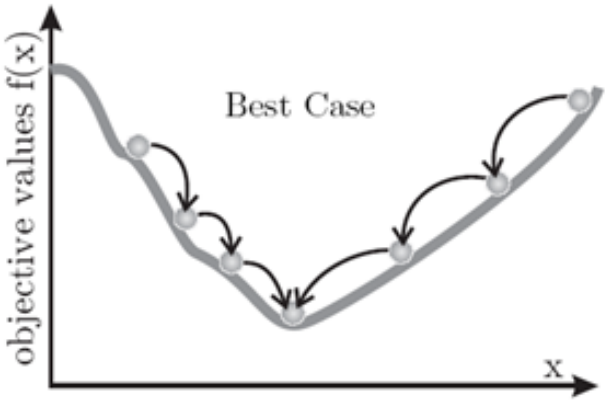


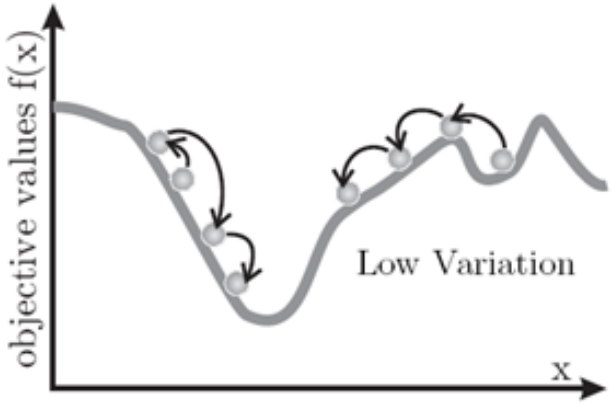
Research model by Mitroff *et al.* (1974)

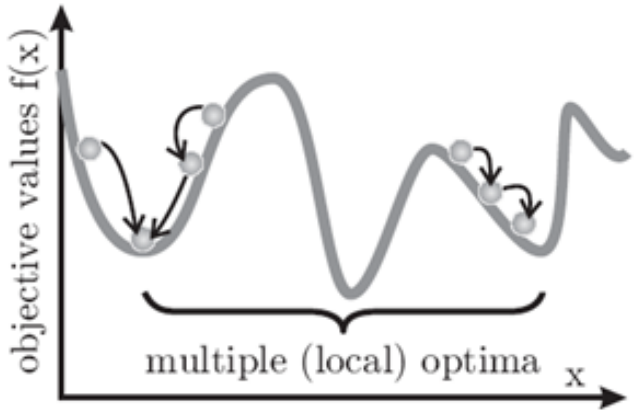


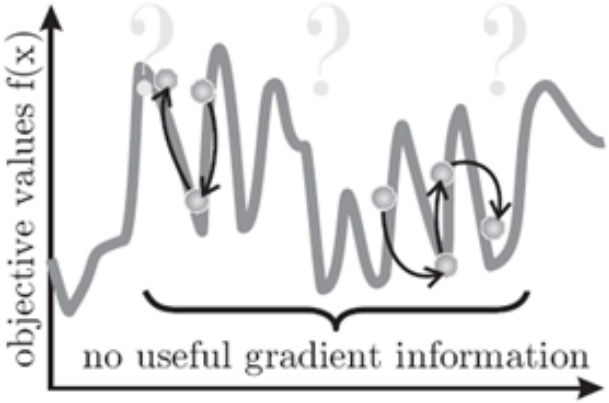
Fonte: TALBI, EL-GHAZALI. Metaheuristics: from design to implementation. John Wiley & Sons, 2009.

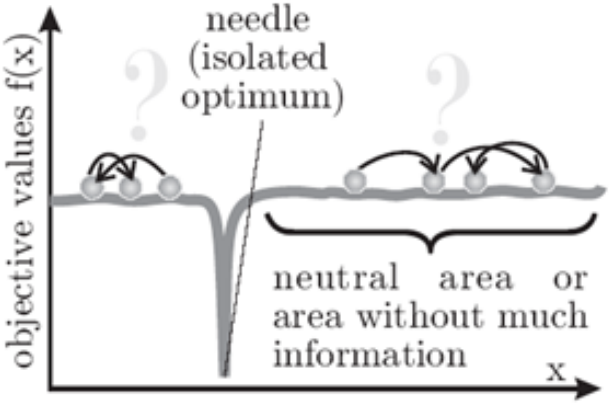
Algumas situações comuns:





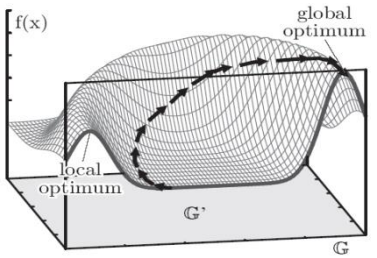






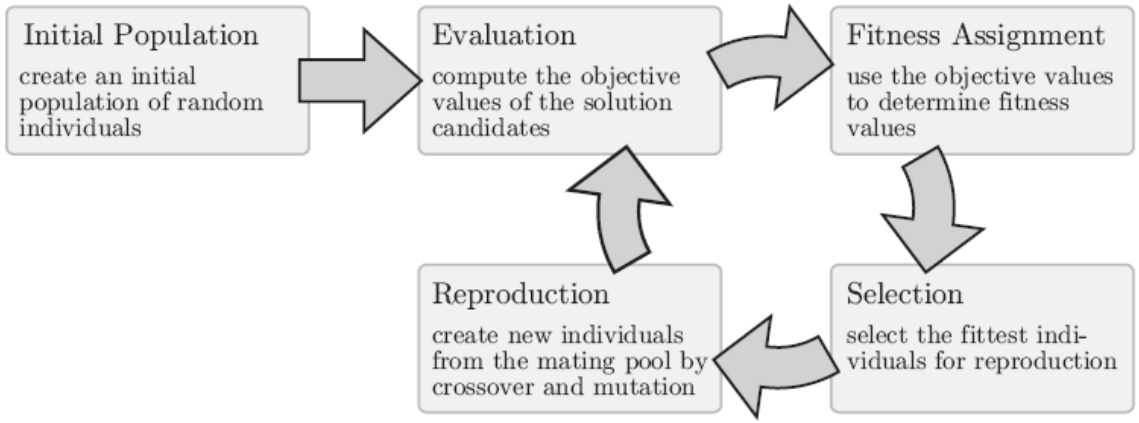
Metaheurísticas:

- **Métodos capazes de sair de ótimos locais permitindo a busca em regiões mais promissoras com um custo computacional razoável.**
- **Tipos de metaheurística:**
  - **Baseadas em um único indivíduo:**



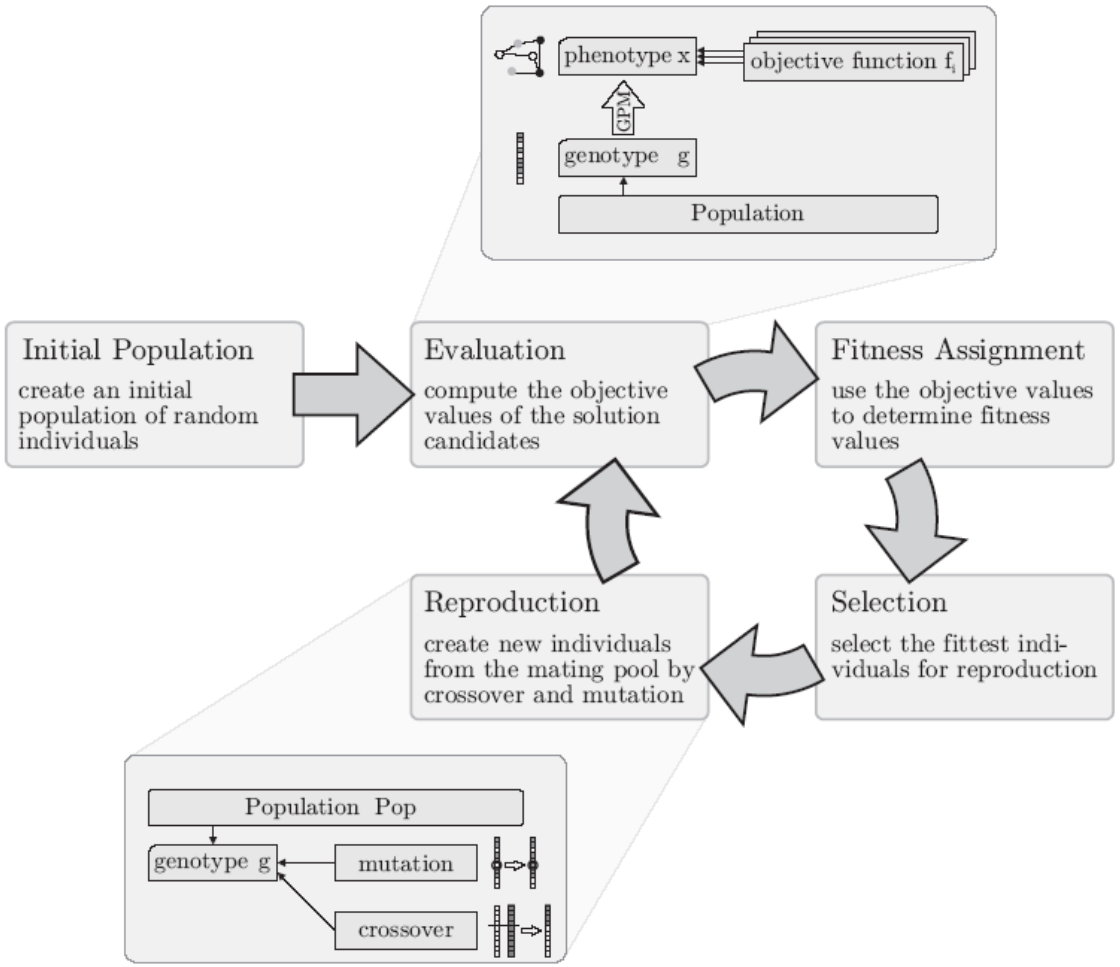
Ex: Busca Tabu, Path-relinking, Simulated Annealing

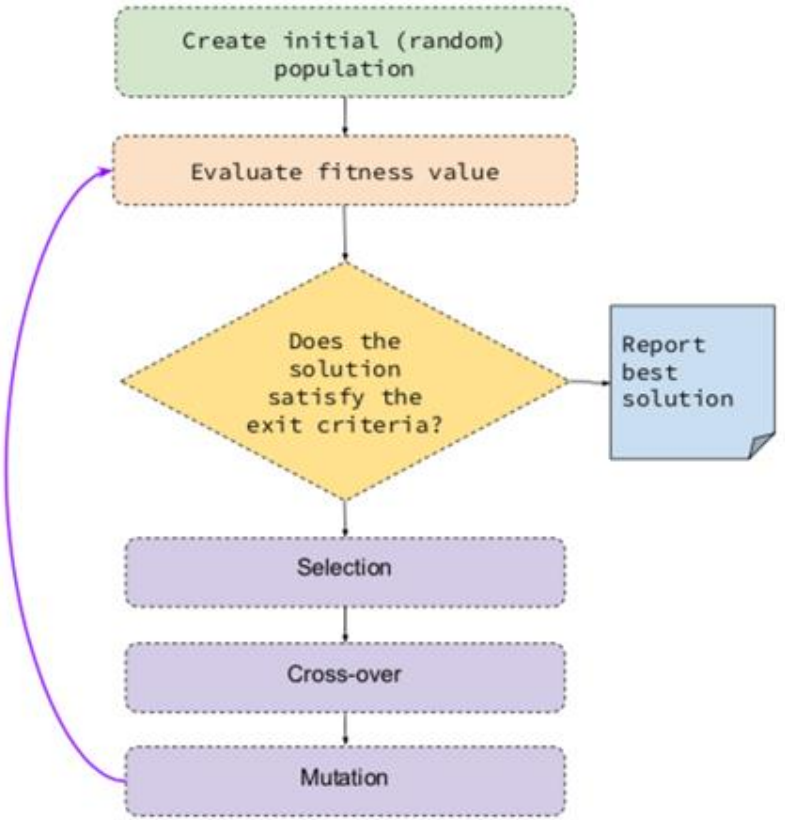
  - **Baseadas em uma população de indivíduos:**



Algoritmo genético:

- **É uma metaheurística que se baseia nos princípios da seleção natural:**





Algoritmo genético:

- Formas de representação das soluções (uso de cromossomos):
  - Binária: (Ex: Problema da mochila binário →

Objeto	1	2	3	4	5	6	7	8	9	10
	0	1	0	1	1	0	0	1	0	0

)
  - Permutação: (Ex: TSP →

Cidades	1	6	3	5	9	7	2	10	8	4

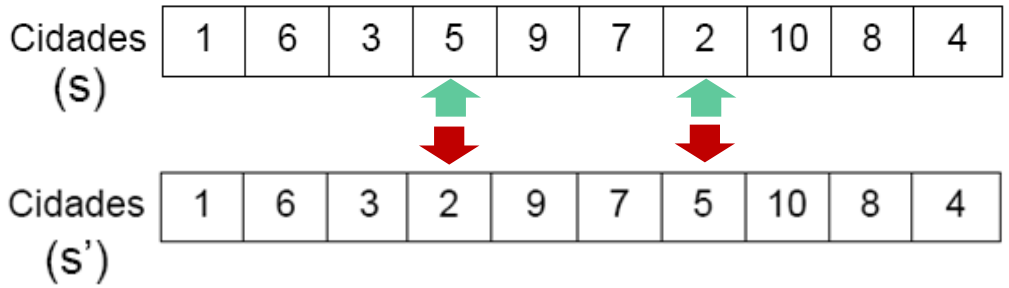
)
  - Número contínuo [min,max]:

0	1	0	1	1	0	0	1	0

$$x = \min + (\max - \min)\frac{b_{10}}{2^{9}-1}$$

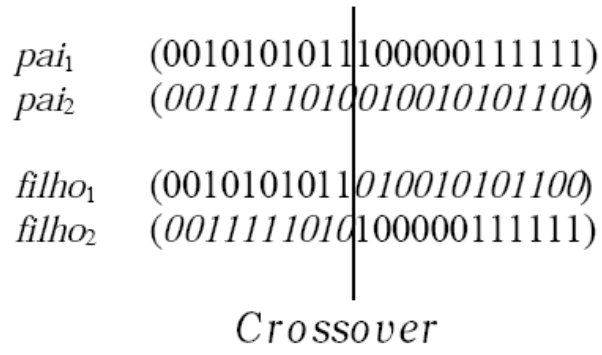
- Vizinhança: um vizinho s' de uma solução s é uma solução na qual foi aplicado um movimento (definido a priori) modificando a solução corrente.

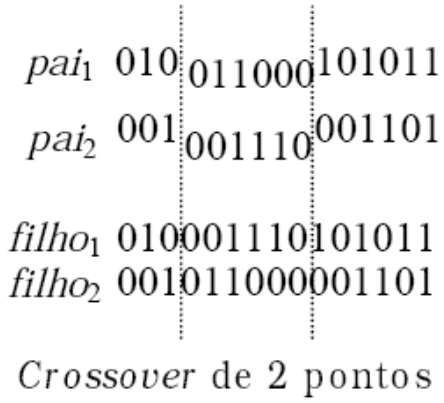
Exemplo:

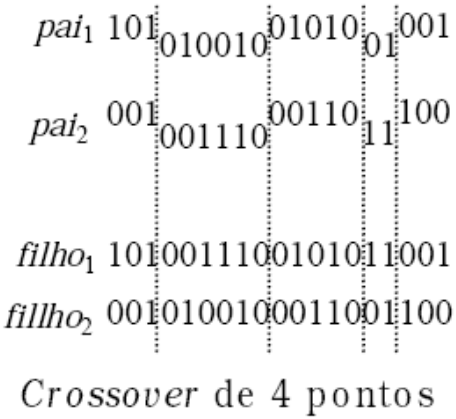


Algoritmo genético:

- Operadores: crossover e mutação
  - Crossover:







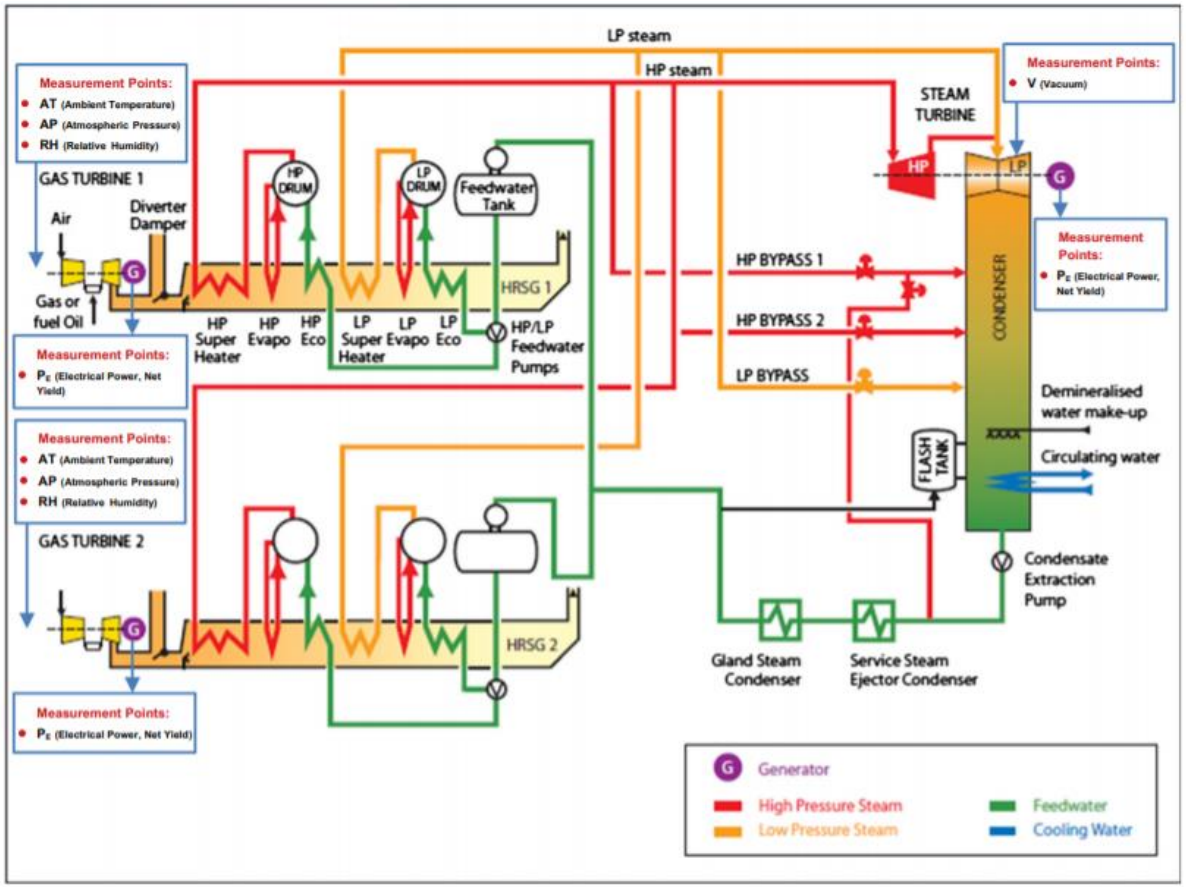
  - Mutação:

Antes
filho$_1$ (0010101010010010101100)
filho$_2$ (0011111011100000111111)

Depois
filho$_1$ (0010**0**0101001001011100)
filho$_2$ (0011111011**0**00000111111)

Otimização (Prescriptive Analytics):

- **Produção da energia gerada (PE):**



Parâmetros:

AT : Ambient Temperature

V: Exhaust Vacuum Speed

AP: Atmosferic Pressure

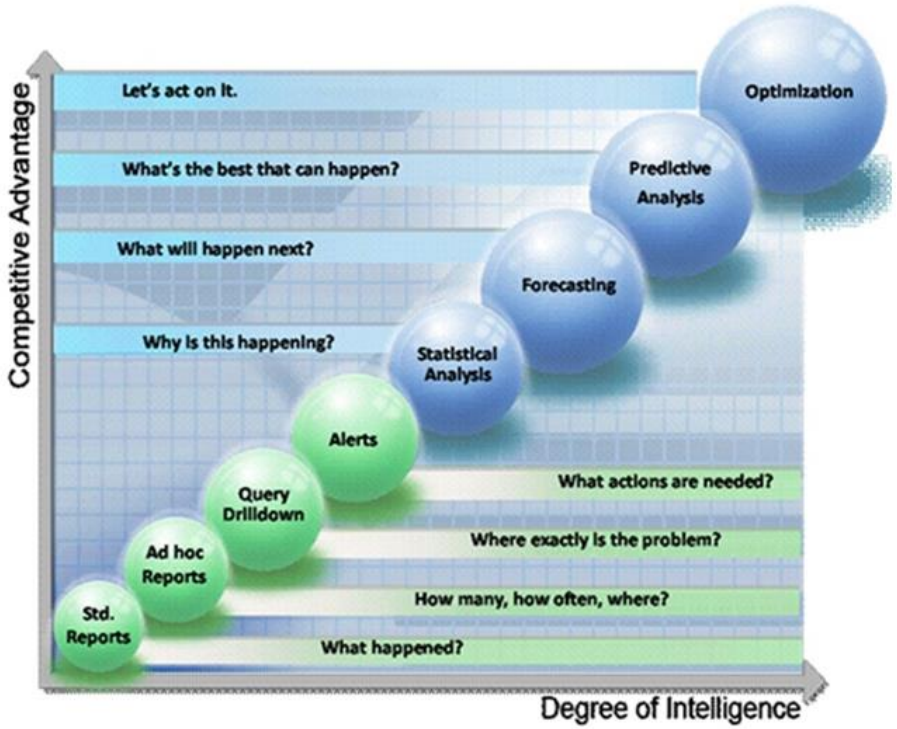
RH: Relative Humidity

Função a ser otimizada:

Modelo de regressão (previsão de PE) → opção selecionada: **Random Forest** (maior R^2 e menor RMSE)

Otimização (Prescriptive Analytics):

- Previsão de Produção de Energia → Otimização




```
— Genetic Algorithm ———————————————

GA settings:
Type                  =  real-valued
Population size       =  50
Number of generations =  100
Elitism               =  2
Crossover probability =  0.8
Mutation probability  =  0.1
Search domain =
         x1    x2     x3     x4
lower  1.81 25.36  992.9  25.56
upper 37.11 81.56 1033.3 100.16

GA results:
Iterations             = 100
Fitness function value = 494.4157
Solution =
          x1       x2       x3       x4
[1,] 5.10924 40.07584 1012.263 62.99484
```

Solução obtida:

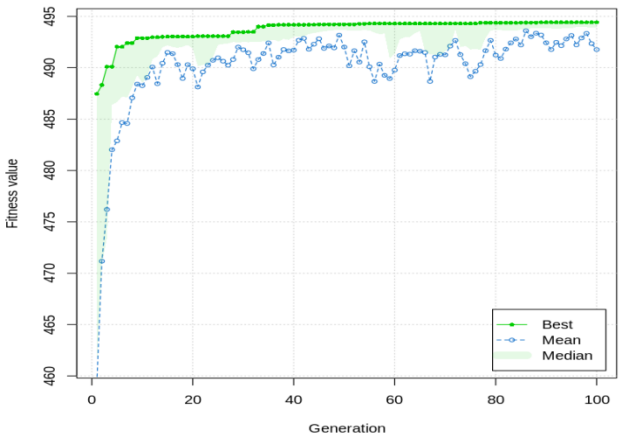
FO=494.42

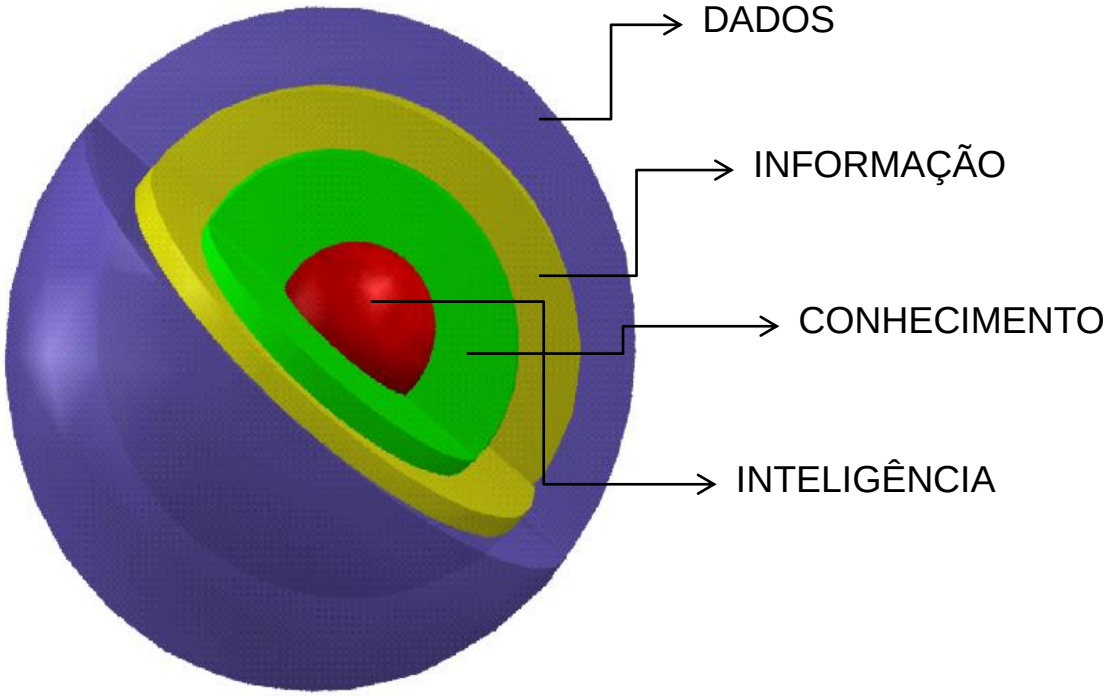
AT=5.11

V=40.08

AP=1012.26

RH=62.99





Rodrigo A. Scarpel

rodrigo@ita.br

http: www.ief.ita.br/~rodrigo